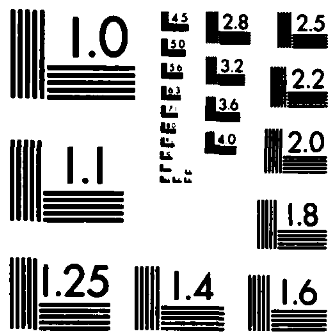
AD-A147 766 OPTIMUM JAMMING EFFECTS ON FREQUENCY-HOPPING M-ARY FSK (FREQUENCY-SHIFT K..(U) LEE (J S) ASSOCIATES INC ARLINGTON VA J S LEE ET AL. OCT 84 JC-2025-N 1/7

UNCLASSIFIED N00014-83-C-0312 F/G 17/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

JC-2025-N

AD-A147 766

OPTIMUM JAMMING EFFECTS ON FREQUENCY-HOPPING M-ARY FSK SYSTEMS UNDER CERTAIN ECCM RECEIVER DESIGN STRATEGIES

PREPARED FOR:

THE OFFICE OF NAVAL RESEARCH
STATISTICS AND PROBABILITY PROGRAM
ARLINGTON, VIRGINIA 22217

FINAL REPORT
UNDER CONTRACT N00014-83-C-0312
(NR 042-447)

DTIC FILE COPY

OCTOBER 1984

J. S. LEE ASSOCIATES, INC.
2001 JEFFERSON DAVIS HIGHWAY, SUITE 601
ARLINGTON, VIRGINIA 22202

12

OPTIMUM JAMMING EFFECTS ON FREQUENCY-HOPPING M-ARY FSK SYSTEMS UNDER CERTAIN ECCM RECEIVER DESIGN STRATEGIES

OCTOBER 1984

JC-2025-N

FINAL REPORT

UNDER CONTRACT N00014-83-C-0312

(NR 042-447)

Prepared For:

THE OFFICE OF NAVAL RESEARCH

Statistics and Probability Program

Arlington, Virginia 22217

Prepared By:

J. S. LEE ASSOCIATES, INC.

2001 Jefferson Davis Highway, Suite 601

Arlington, Virginia 22202

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A147766

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
The Optimum Jamming Effects on Frequency-Hopping M-ary FSK Systems Under Certain ECCM Receiver Design Strategies

5. TYPE OF REPORT & PERIOD COVERED
Final Report
March 1, 1984-Sept. 30, 1984

6. PERFORMING ORG. REPORT NUMBER
JC-2025-N

7. AUTHOR(s)
Jhong S. Lee, Leonard E. Miller, Robert H. French, Young K. Kim, Arman P. Kadrichu

8. CONTRACT OR GRANT NUMBER(s)
N00014-83-C-0312

9. PERFORMING ORGANIZATION NAME AND ADDRESS
J. S. Lee Associates, Inc.
2001 Jefferson Davis Highway, Suite 601
Arlington, Virginia 22202

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
NR 042-447

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
Statistics and Probability Program
Arlington, VA 22217

12. REPORT DATE
October 1984

13. NUMBER OF PAGES
572 + xxxii

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Spread-Spectrum Systems, Frequency Hopping, M-ary Frequency-Shift Keying, Jamming, Partial-Band Noise Jamming, Tone Jamming, Bit Error Rate, ECCM, ECM, Diversity Gain

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The performance of M-ary frequency-shift-keying frequency-hopping systems using multiple hops per symbol waveform is derived and evaluated for the cases of worst-case partial-band noise jamming and worst-case partial-band tone jamming. The analyses include the effects of the jamming under certain ECCM receiver design strategies. The receiver design strategies include the conventional square-law linear combiner and the several types of unconventional nonlinear combiners. The ECCM receiver designs employing the nonlinear com-

(Block 20 Cont'd)

biners are found to provide diversity gains whereas such gains are not realized under the conventional linear combining strategy.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Concluded)

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Concluded)

LIST OF TABLES

LIST OF TABLES
(Continued)

1.0 INTRODUCTION

The contractual studies and analyses which are presented in this final report are concerned with the electronic counter-countermeasures (ECCM) performances of several different receiver processing schemes for uncoded frequency-hopped multiple-frequency-shift-keying (FH/MFSK) radio communication systems. The systems which we have considered belong to the class known as fast frequency hopping, in which the M-ary symbol (one of M baseband frequencies representing $K = \log_2 M$ bits of information) is transmitted L times on L successive hops with pseudorandomly selected carrier frequencies. In this manner the energy in the spread-spectrum FH/MFSK waveform symbols is further spread over time, enhancing the low-probability-of-intercept (LPI) qualities of the modulation. The main question addressed by our studies reported herein is whether the use of these L hops per M-ary symbol also provides a form of time-diversity anti-jamming improvement for the system.

As a prelude to the detailed mathematical analyses and graphical summaries of numerical computations in Sections 2 through 8 of this report, the remainder of this section discusses the background of spread-spectrum communications to set the context for these studies. This section then concludes with an executive summary of the important findings and results contained in the body of the report.

The main body of the report is divided into seven sections. Sections 2, 3, 4, and 5 present details of the analysis of the effects of partial-band noise jamming on the performance of the square-law linear combining receiver, the adaptive gain control receiver, the clipper receiver, and the self-normalizing receiver,* respectively. In section 6 we compare the performance of

*These receiver types are further defined in Section 1.3.1.

the different receiver structures and discuss how the results of our work can be applied to design of an ECCM radio system. In Section 7 we change sides, and take the jammer's viewpoint to discuss how the results of this study can be applied to the problem of optimizing the design of a jamming system. Section 8 considers the topic of tone jamming and its effects on the performance of the square-law linear combining receiver. The report concludes with a number of appendices which contain mathematical proofs and derivations in support of the main text, as well as listings of the major computer programs used in obtaining numerical results. Specific comments applicable to all of these computer program listings are contained in Appendix 1A to avoid unnecessary, tedious repetition.

1.1 BACKGROUND

A fundamental requirement of modern military communications is to achieve reliable transmission of signals over a channel that is affected by interference of several types. When the interference is generated by a hostile party with the intention of disrupting the communications link, the channel is subjected to an electronic warfare environment.

In an electronic warfare environment, the communicator and the interferer (the jammer) both have an uncompromising conflicting interest to achieve their own respective objectives. The objective of the communicator is to design the communication system to render a low probability of intercept (LPI); and further, if jammed as a result of being detected (intercepted), to mitigate the effects of the interference. To an interceptor, on the other hand, the primary objective is to optimize the use of his available jamming power to victimize the communicator's LPI signal traffic to the desired level of degradation.

Consider first the objectives of the communicator, who seeks to achieve his design goals for the system by providing anti-jam (or jam resistant) features as well as an inherently high processing gain for reducing the performance-degrading influence of jamming power in the demodulation process. When the communicator realizes that the jammer's strategy is to optimize the application of his jamming power resource upon detection of the communicator's signal, the first and foremost objective of the communicator is the design of the most power-efficient LPI waveform to reduce the detectability of his signal. The basic axiom for designing such a waveform is that the energy spectral density must be weak, which leads to the choice of a spread-spectrum waveform.

There are two basic classes of spread-spectrum systems which are commonly used: direct sequence (DS) and frequency hop (FH). In a DS system, a high-rate pseudo-random code is modulo-2 added to the baseband data before it modulates the carrier. The carrier modulation for a DS system is usually some form of phase-shift keying (PSK). In an FH system, a pseudo-random code is used to select one of many available carrier frequencies for transmission at a given time; the information is usually modulated onto the hopped carrier by frequency-shift keying (FSK).

Both DS and FH systems are viable spread-spectrum techniques. However, in an electronic warfare environment where strong jamming is expected, frequency-hopping frequency-shift-keying (FSK) schemes are usually employed for two practical reasons: (1) the frequency-hopping system is capable of providing high processing gain; and (2) the FSK modulation can be processed noncoherently. A DS-SS system has two drawbacks: (1) current technology does not allow the generation of pseudo-random sequences at rates higher than a few hundred megachips per second, which may not provide sufficient processing gain; and (2) the DS-SS system necessarily employs phase-shift-keying modulation

and, hence, phase coherence over the spreading bandwidth is required. Current technology is limited to only 50 MHz or less for achieving phase coherence.

Given that the system is to employ frequency hopping, the data modulation will be some form of MFSK. It is well known that the use of higher-order alphabets with conventional (non-hopped) M-ary FSK (MFSK) gives substantial performance improvement for equal bit energy-to-thermal noise density ratios in the Gaussian channel, as shown in Figure 1-1. Thus, it is natural to consider the use of MFSK in conjunction with FH in a spread-spectrum system. However, the performance of the FH/MFSK system under jamming conditions must be known before such a system design decision can be made. Similarly, the performance of alternative receiver structures must be available to the system designer in order to make an appropriate selection.

We now turn our attention to the interceptor's point of view. His objective is to detect the communicator's LPI signal and, upon detection, initiate jamming operations. As illustrated in Figures 1-2 and 1-3, the jammer may be airborne and hence jamming power is limited. Thus, the jammer's obective is to maximize the degradation of the victim's communication link with minimum application of his jamming power. It is well known to both the communicator and the jammer that the optimum way of utilizing jamming power against FH-SS systems is to concentrate the (limited) jamming power over a selected fraction of the total system bandwidth, which is assumed to be known to the jammer. This strategy is called partial-band jamming.

1.2 OBJECTIVE OF THIS STUDY

The purpose of this report is to study the optimum strategies available to both the communicator and the interceptor under existing constraints placed by each party. To a communicator, the optimum selection of the number of hops per symbol under the worst-case partial-band-jamming environment is the primary

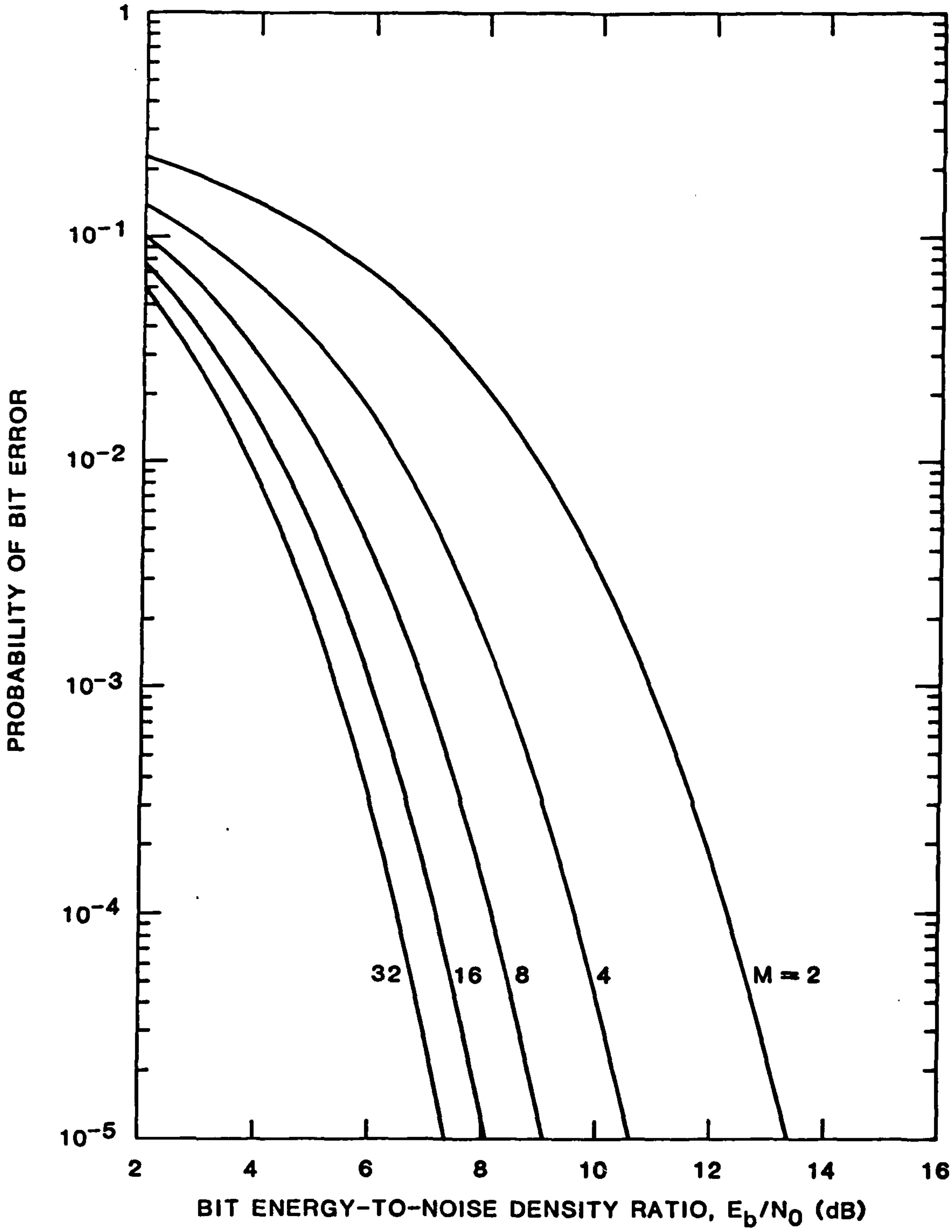


FIGURE 1-1 PROBABILITY OF BIT ERROR VS. BIT ENERGY-TO-NOISE DENSITY RATIO FOR MFSK

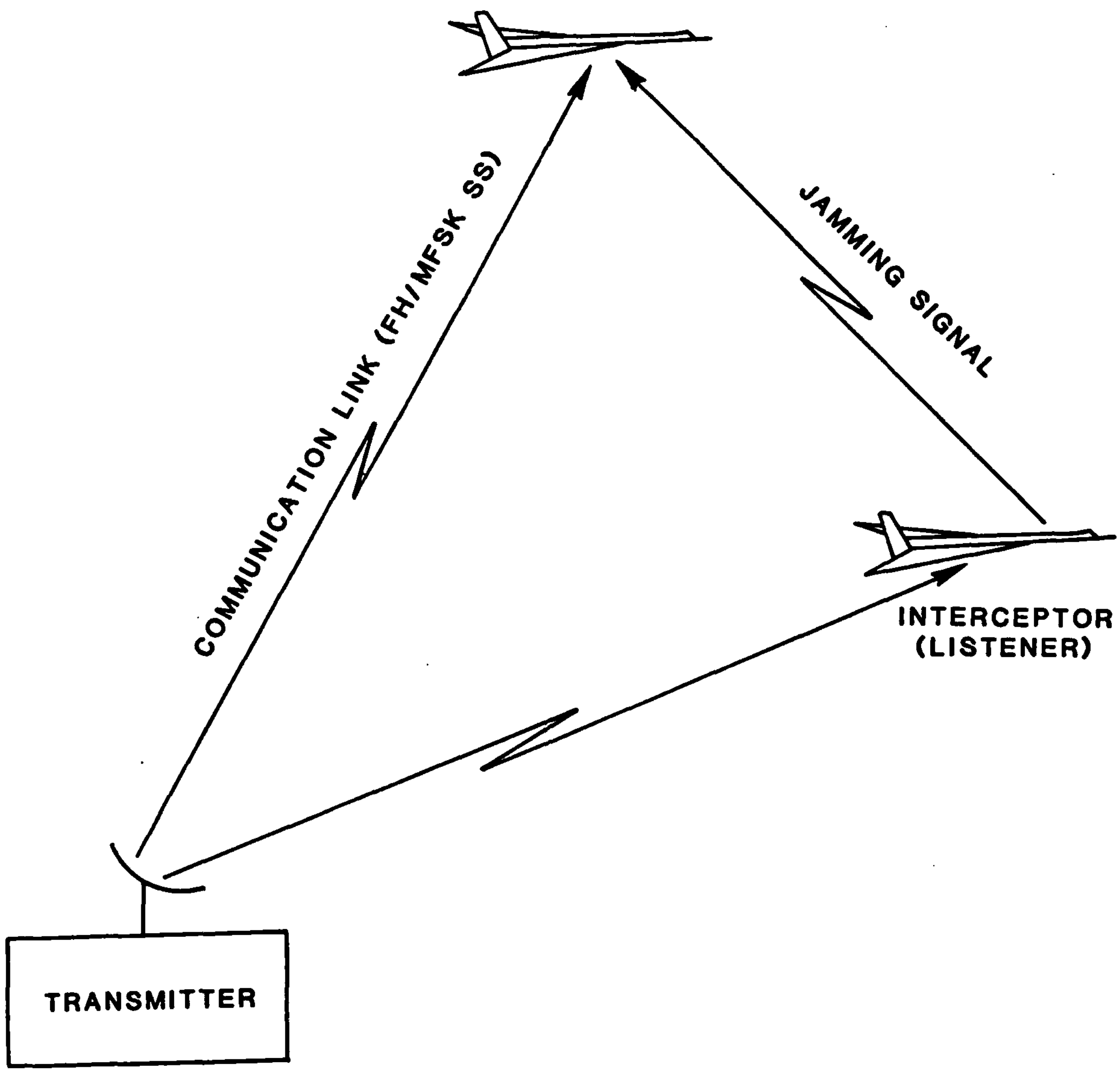


FIGURE 1-2 FREQUENCY HOPPING MFSK SPREAD-SPECTRUM LPI COMMUNICATION IN EW ENVIRONMENT (GROUND-TO-AIR COMMUNICATION LINK)

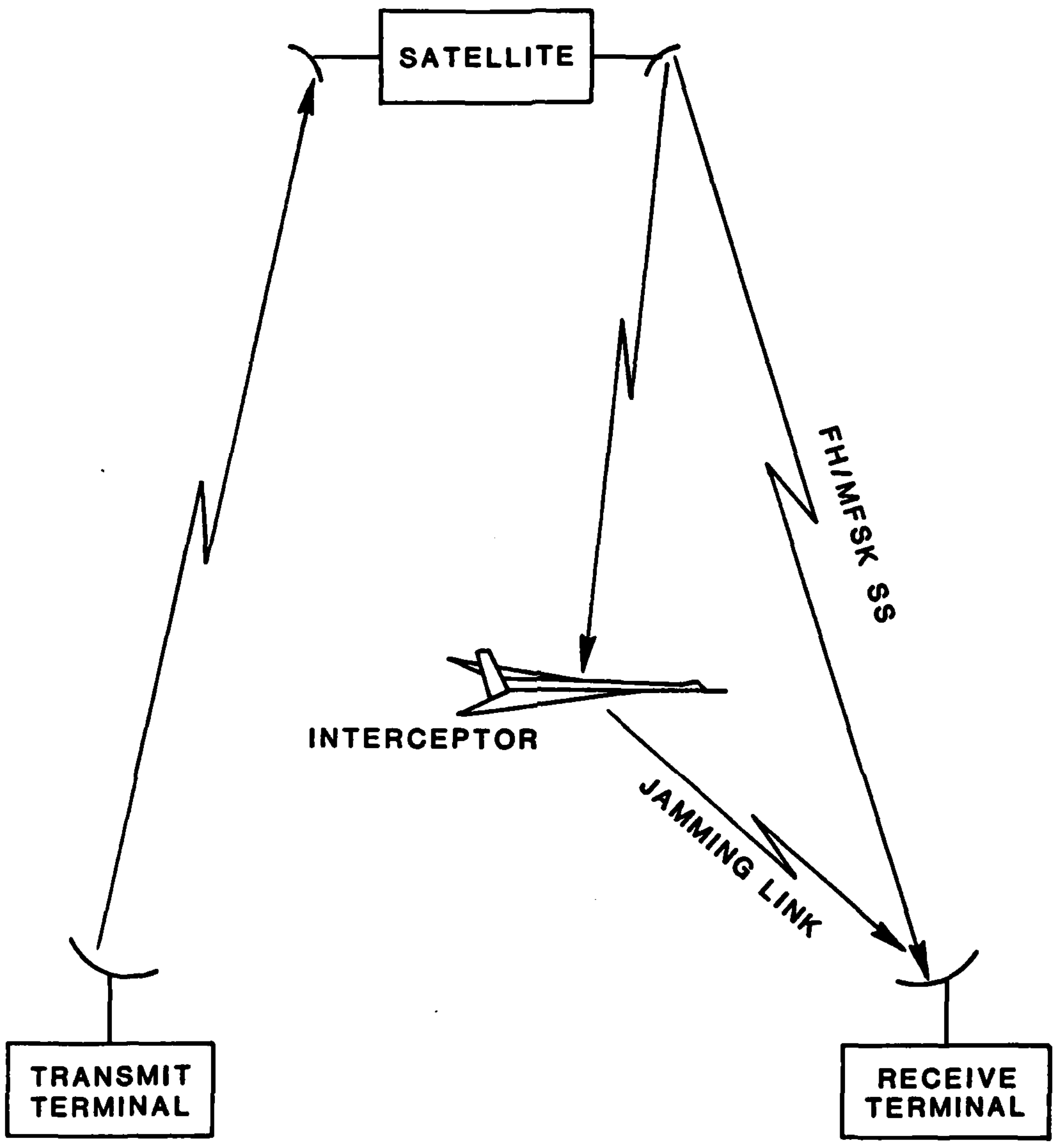


FIGURE 1-3 FREQUENCY HOPPING MFSK SPREAD-SPECTRUM LPI COMMUNICATION IN EW ENVIRONMENT (GROUND-SATELLITE-GROUND COMMUNICATION LINK)

knowledge he must possess. He must recognize that when the ratio of jamming power to his communication transmitter power exceeds a certain level, a severe constraint is placed on the choice of L (the number of hops per symbol) if the message bit stream is to be recovered with an acceptable error probability.

In our previous work [1], we considered the performance of FH/BFSK (binary) systems in the partial-band noise-jamming channel and in the tone-jamming channel. We also considered several receiver structures including both linear and nonlinear diversity combining structures. One important result of this work was the observation that the effects of thermal noise must be included in the analysis if misleading results are to be avoided.

The present work extends the prior work to the case of FH/MFSK systems in the partial-band noise-jamming channel and in the tone-jamming channel. Past efforts at analysis of M-ary systems have frequently avoided the analytical and numerical difficulties associated with an exact analysis by resorting to approximations such as the union bound. However, we have discovered that the use of the union bound can introduce substantial errors which are unacceptable. Indeed, the behavior can be so anomalous as to go contrary to what is expected. As shown in Figure 1-4, the application of the union bound to FH/MFSK on the partial-band noise-jamming channel would lead one to conclude that increasing M would degrade performance, which is counter to the expected improvement due to M-ary coding gain. This same anomalous behavior of the union bound was observed by Crepeau and McGregor [12] when thermal noise was neglected in the analysis. Given this behavior of the approximate analysis, we have no choice in this study but to use exact formulations of the bit error probability.

1.3 SUMMARY OF RESULTS

We consider for analysis a communications system in which the source sequence of binary digits at rate $R_b = 1/T_b$ is encoded into M-ary symbols,

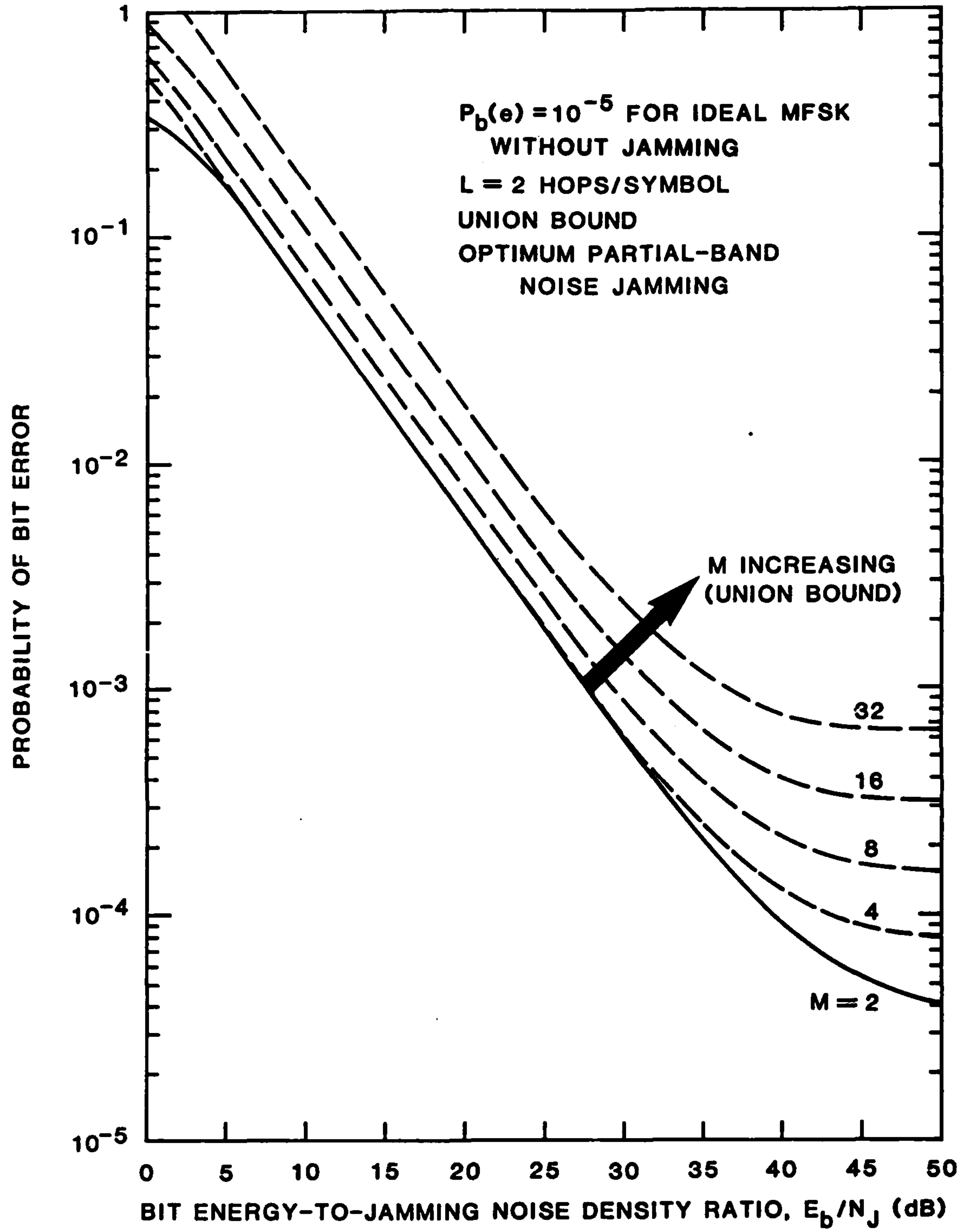


FIGURE 1-4 BIT ERROR PROBABILITY USING UNION BOUND FOR SQUARE-LAW LINEAR COMBINING RECEIVER

with $M = 2^K$, at a symbol rate of $R_s = R_b/K = 1/T_s$. As shown in Figure 1-5, these M-ary symbols are applied to a baseband MFSK modulator which selects one of M signaling frequencies. The output of the modulator is mixed with a hopping local oscillator which hops, under control of a pseudo-random code generator, at a rate $R_H = 1/\tau$. The output of this mixer is passed through a bandpass filter with bandwidth equal to the system bandwidth, up-converted to the final radio frequency, amplified, and radiated from the antenna. The final hopping waveform is illustrated in the frequency-time diagram shown in Figure 1-6.

1.3.1 Results for Partial-Band Noise Jamming

The rationale for using the multiple hops per symbol is to counter the effect of intentional jamming. From the viewpoint of the jammer, wideband noise jamming is the least effective strategy. Instead, the jammer may employ a partial-band noise-jamming strategy, as illustrated in Figure 1-7, wherein only a fraction γ of the system bandwidth W is jammed with Gaussian noise.

We have analyzed the performance of several receivers in the partial-band noise-jamming channel. In conducting these analyses, we have included the effects of thermal noise and have used exact formulations for the error rates rather than bounding techniques. This latter point is quite important since there are known instances where the union bound gives anomalous behavior, as discussed in Section 1.2. As our results show, the partial-band noise-jamming channel is, in fact, one of the cases where the union bound sometimes fails.

All of the receiver structures which we have analyzed can be summarized by the block diagram shown in Figure 1-8 and the specifications in Table 1-1. The four receivers analyzed are the linear combining receiver, the clipper receiver, the adaptive gain control (AGC) receiver, and the self-

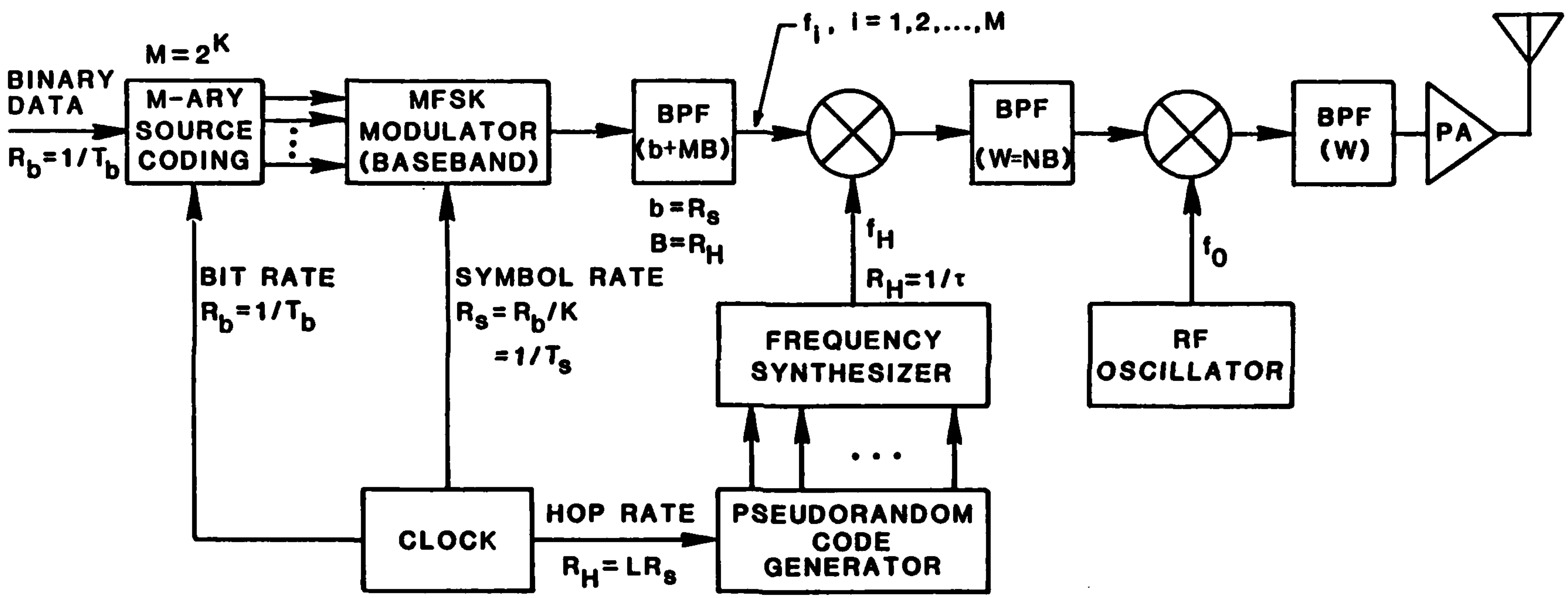


FIGURE 1-5 L HOPS/SYMBOL FH/MFSK TRANSMITTER

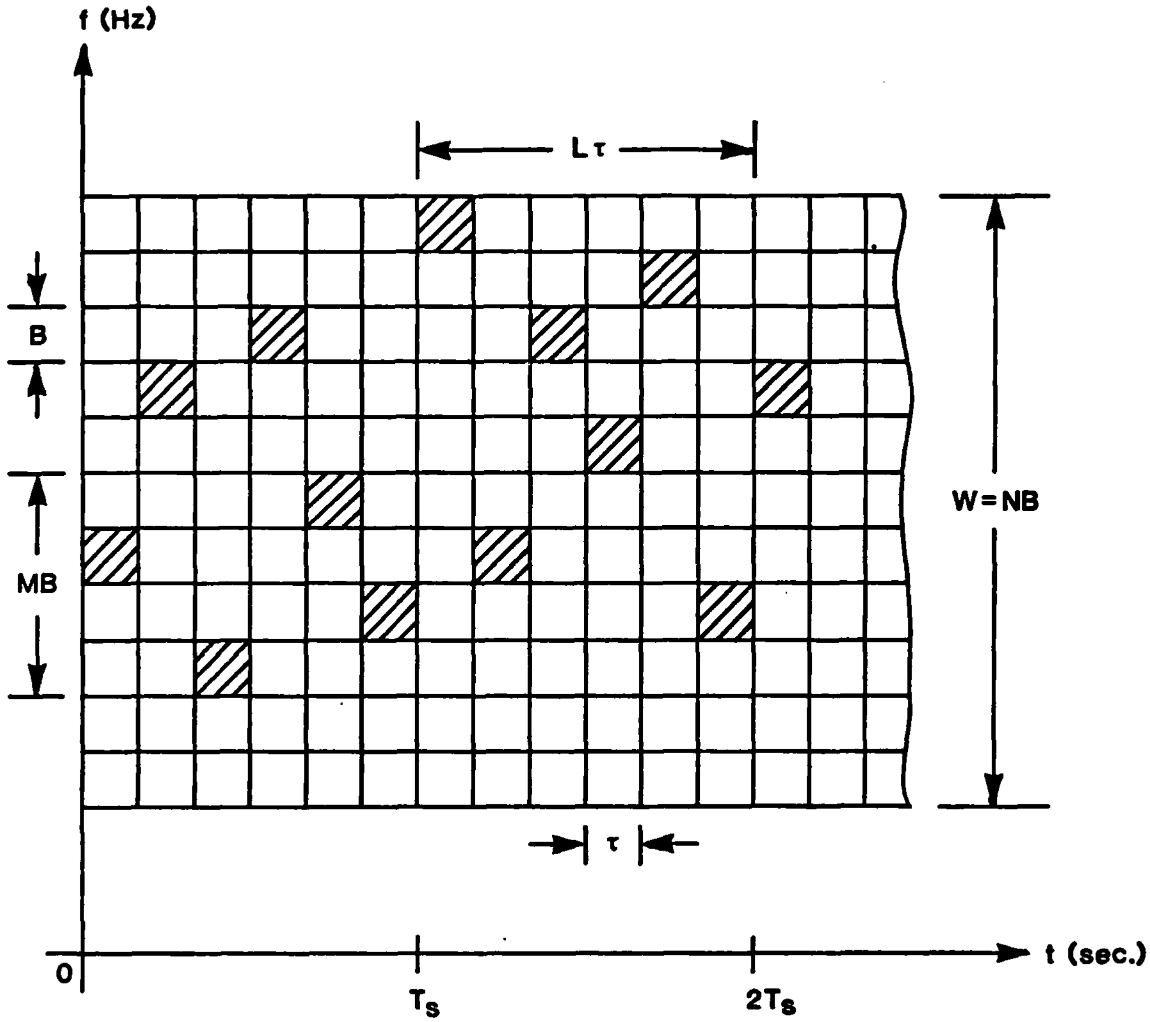


FIGURE 1-6 TYPICAL L HOPS/SYMBOL FH/MFSK WAVEFORM PATTERN

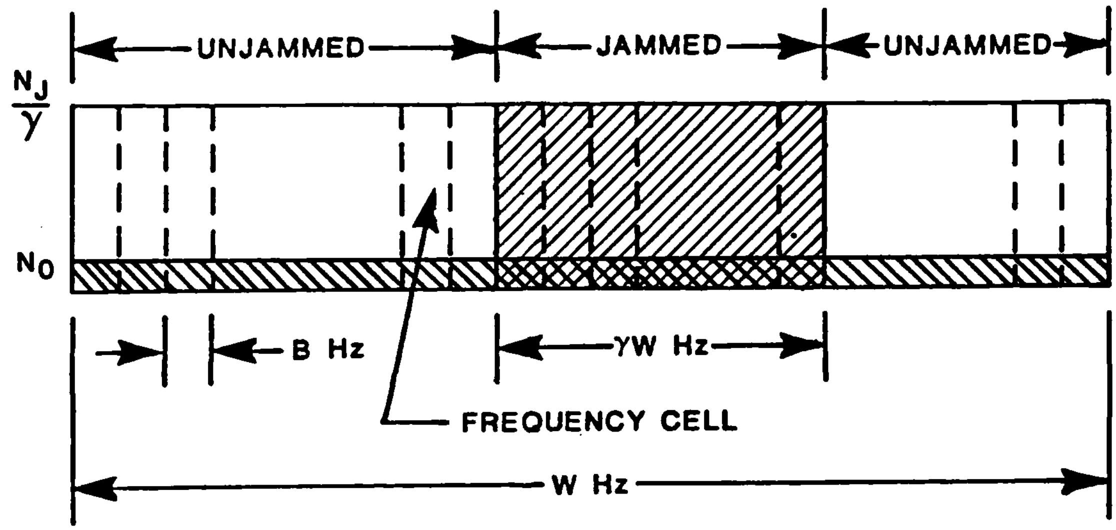


FIGURE 1-7 THERMAL NOISE AND PARTIAL-BAND NOISE JAMMING MODEL

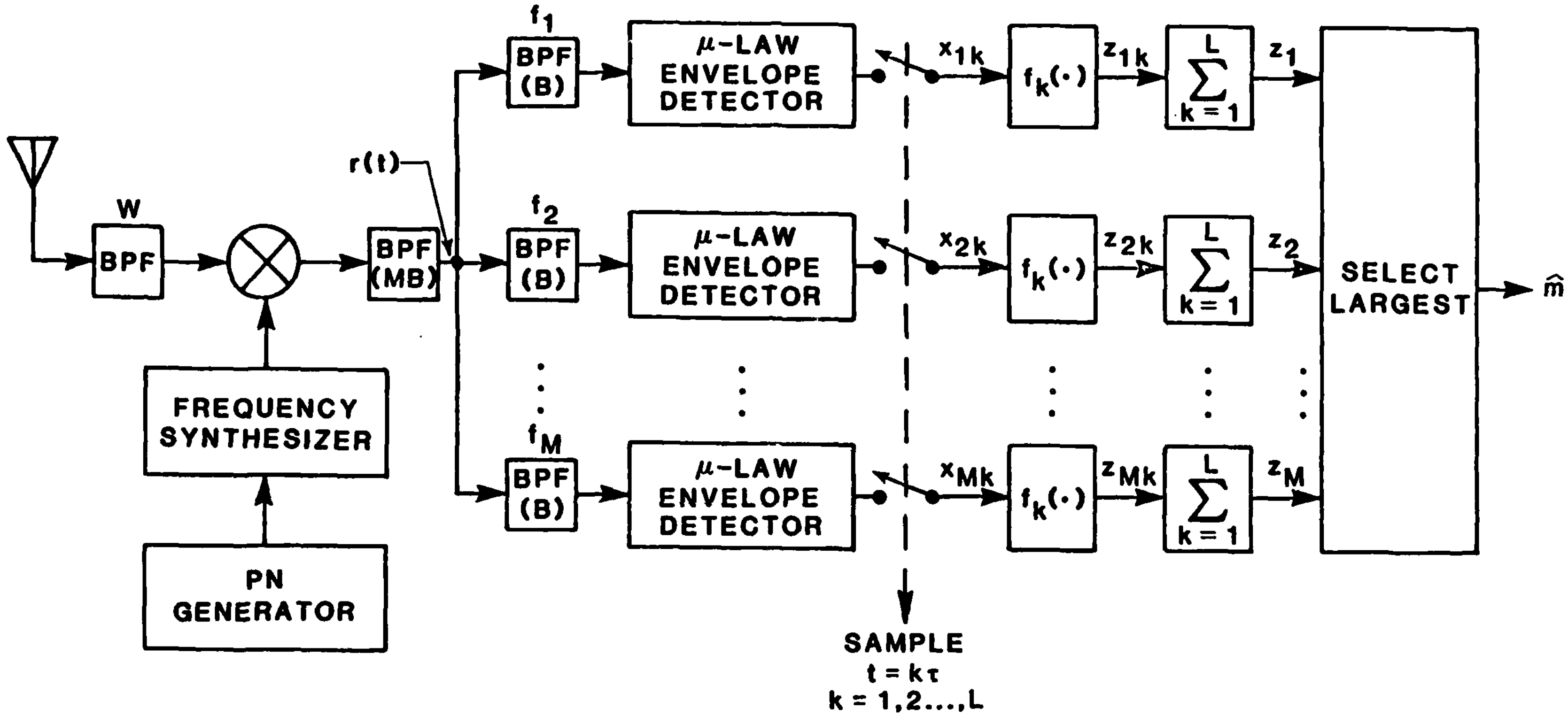


FIGURE 1-8 FH/MFSK RECEIVER STRUCTURE

TABLE 1-1

DESCRIPTIONS OF THE RECEIVERS

RECEIVER TYPE	SPECIFICATION OF $z_{ik} = f_k(x_{ik}), i=1,2,\ldots,M$	REMARKS	IS SIDE INFORMATION ON JAMMING STATE USED IN DECISION?
LINEAR COMBINING RECEIVER	$z_{ik} = x_{ik}$	Direct Connection (Linear Combining)	No
CLIPPER RECEIVER	$z_{ik} = \begin{cases} x_{ik}, & x_{ik} \leq \eta \\ \eta, & x_{ik} > \eta \end{cases}$	Soft Limiter (Nonlinear Combining)	No
AGC RECEIVER	$z_{ik} = x_{ik}/\sigma_k^2$ $\sigma_k^2 = \begin{cases} \sigma_N^2, & \text{if not jammed} \\ \sigma_N^2 + \sigma_J^2, & \text{if jammed} \end{cases}$ (σ_k^2 = measured)	Adaptive Gain Control (Nonlinear Combining)	No
SELF-NORMALIZING RECEIVER	$z_{ik} = \dfrac{x_{ik}}{\sum_{i=1}^{M} x_{ik}}$	Practical Realization of AGC Using In-Band Measurements	No

normalizing receiver. The linear combining receiver is a conventional design approximation based on a maximum-likelihood receiver for the Gaussian channel. The clipper receiver inserts soft limiters to restrict the ability of jammed hops to dominate the decision process. The AGC receiver uses an additional channel to measure the noise level to provide hop-by-hop normalization of the detector outputs. Finally, the self-normalizing receiver is a practical adaptation of the AGC receiver using in-band measurements to perform the hop-by-hop normalization of the detector outputs.

In addition to the characterization of the receivers by the weighting of the hop-by-hop samples, we may also characterize them by the power-law characteristic of the envelope detector. In Figure 1-8, we show a μ-law envelope detector. If $\mu = 1$, we have a linear-law envelope detector, whereas if $\mu = 2$, we have a square-law envelope detector.

The performance of a square-law linear combining receiver is typified by the curves shown in Figure 1-9 for $L = 2$ hops/symbol with the alphabet size, M, as a parameter. Comparison of this figure with Figure 1-4 clearly shows the necessity of performing an exact analysis rather than using the union bound.

Performance comparisons among the several receivers are given in Figures 1-10 and 1-11 for $L = 1$ and $L = 2$ hops/symbol, respectively. Additional performance curves may be found in Sections 2 through 5 of this report. We see from the figures that the AGC receiver is uniformly better than the square-law linear combining receiver and the clipper receiver in the partial-band noise-jamming channel.

1.3.2 Results for Partial-Band Tone Jamming

As an alternative to partial-band noise jamming, the jammer may

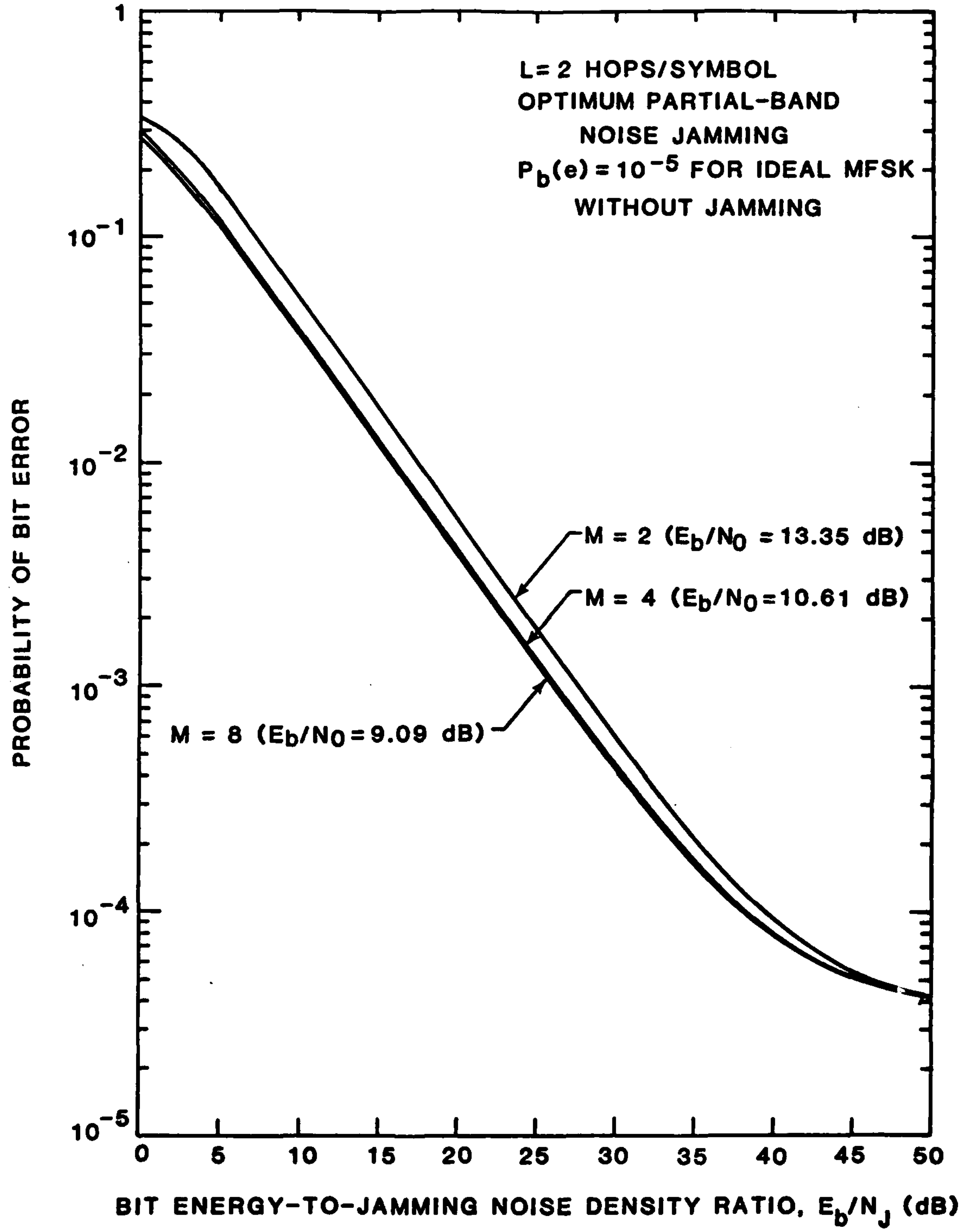


FIGURE 1-9 PROBABILITY OF ERROR VS. E_b/N_J WHEN L=2 AND E_b/N_0 IS SUCH THAT $P_b(e) = 10^{-5}$ FOR IDEAL MFSK (M AS PARAMETER) WITH SQUARE-LAW LINEAR COMBINING RECEIVER

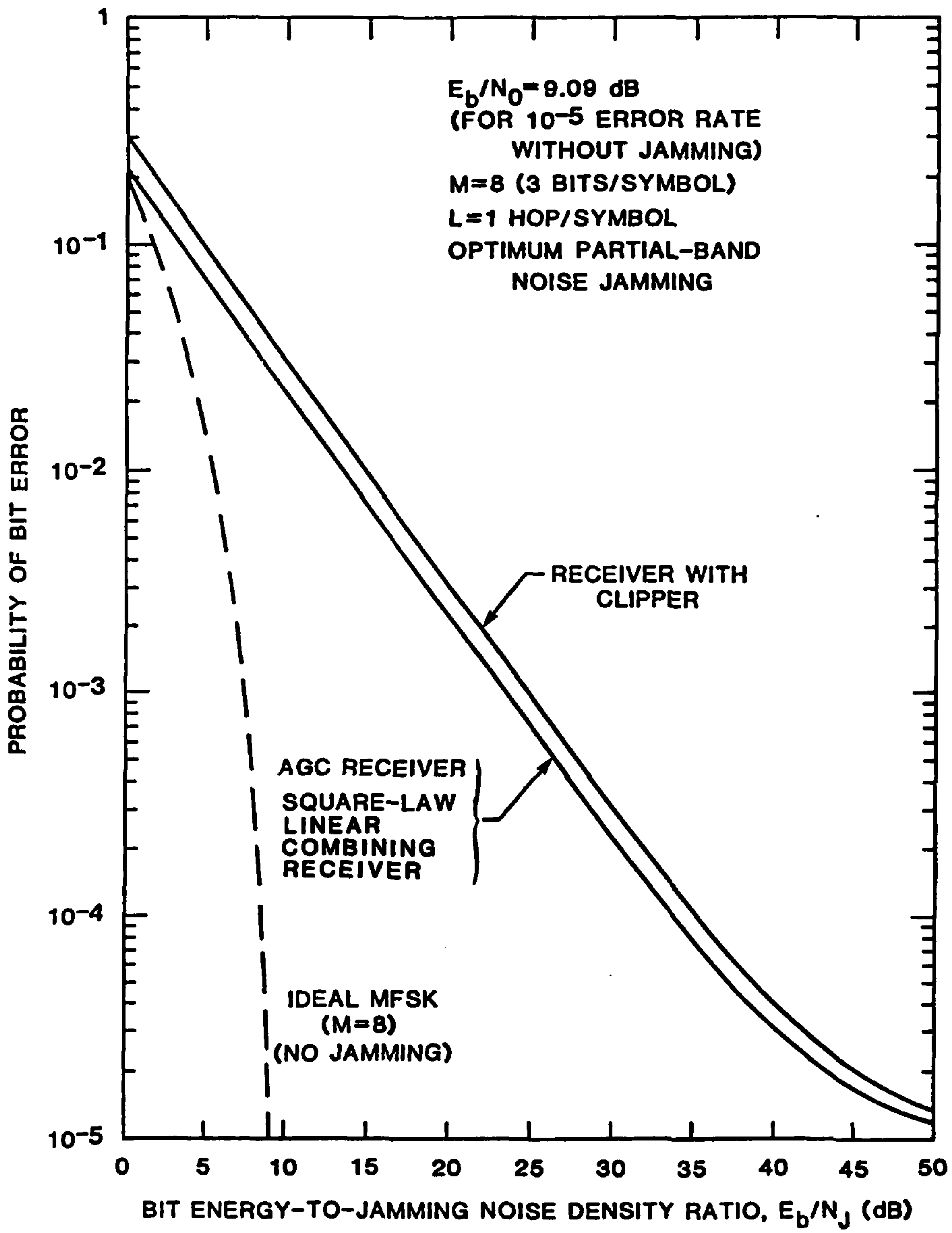


FIGURE 1-10 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/MFSK (M=8) SQUARE-LAW COMBINING RECEIVERS FOR L = 1 HOP/SYMBOL WHEN E_b/N_0= 9.09 dB (FOR IDEAL MFSK (M=8) CURVE THE ABSCISSA READS E_b/N_0)

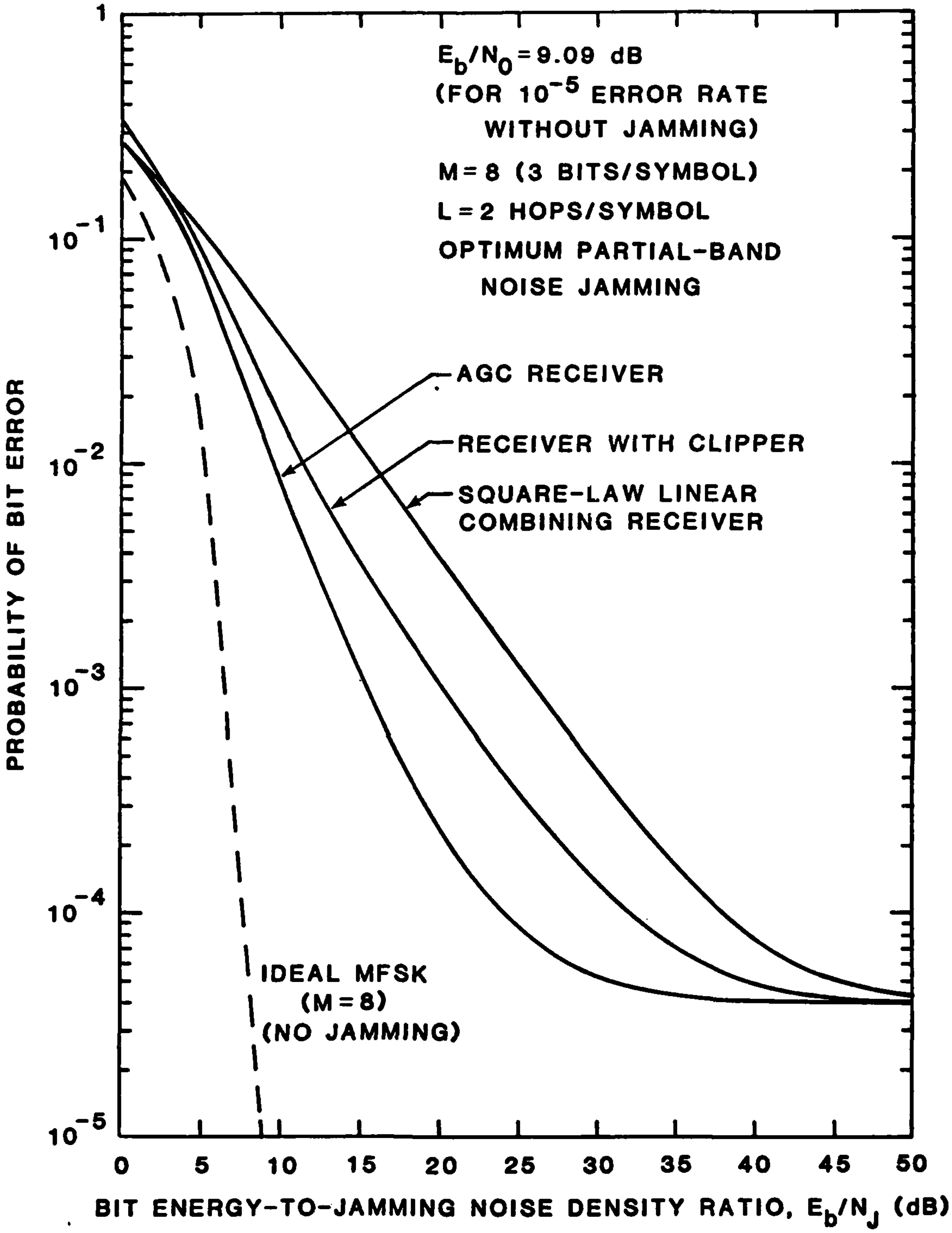


FIGURE 1-11 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/MFSK (M= 8) SQUARE-LAW COMBINING RECEIVERS FOR L= 2 HOPS/SYMBOL WHEN E_b/N_0= 9.09 dB (FOR IDEAL MFSK (M= 8) CURVE THE ABSCISSA READS E_b/N_0)

employ a number of discrete tones. We assume that the jammer will not be able to place a tone in every one of the N available frequency cells within the system bandwidth W, but rather will be able to generate only $q < N$ tones. Hence, we use the term partial-band tone jamming to describe this situation.

The modelling of a partial-band tone jammer is more complicated than the modelling of a partial-band noise jammer. Since the jamming tones are discrete sinusoids, we have the additional parameter of the method by which the q tones are distributed over the N frequency cells. We assume that the total available jamming power J is divided equally among the q tones with each tone having power $J_0 = J/q$. We further assume that there will be at most one jamming tone in any given frequency cell, and that the jamming tone (when present) is at the center of the cell (no frequency difference between the jammer and the signal tones).

We have considered three different partial-band tone jamming models:

- randomly placed tones
- evenly spaced tones (barrage jamming)
- band multitone jamming.

These three models are illustrated in Figure 1-12.

The first model is randomly placed tones in which the jammer makes equiprobable random selections, without replacement, from the N slots to determine where to place his q tones, $1 \leq q \leq N$. This model has also been called independent multitone jamming by some authors [13].

The second model, which we call barrage jamming, consists of q tones spaced at uniform increments of nB Hz; only the starting location for the first tone is picked at random. The maximum number of jamming tones under this model is $q_{max} = N/n + 1$. Furthermore, the largest useful spacing is $n = M$; for if $n \geq M$, then at most one cell of the MB-Hz wide M-ary cluster

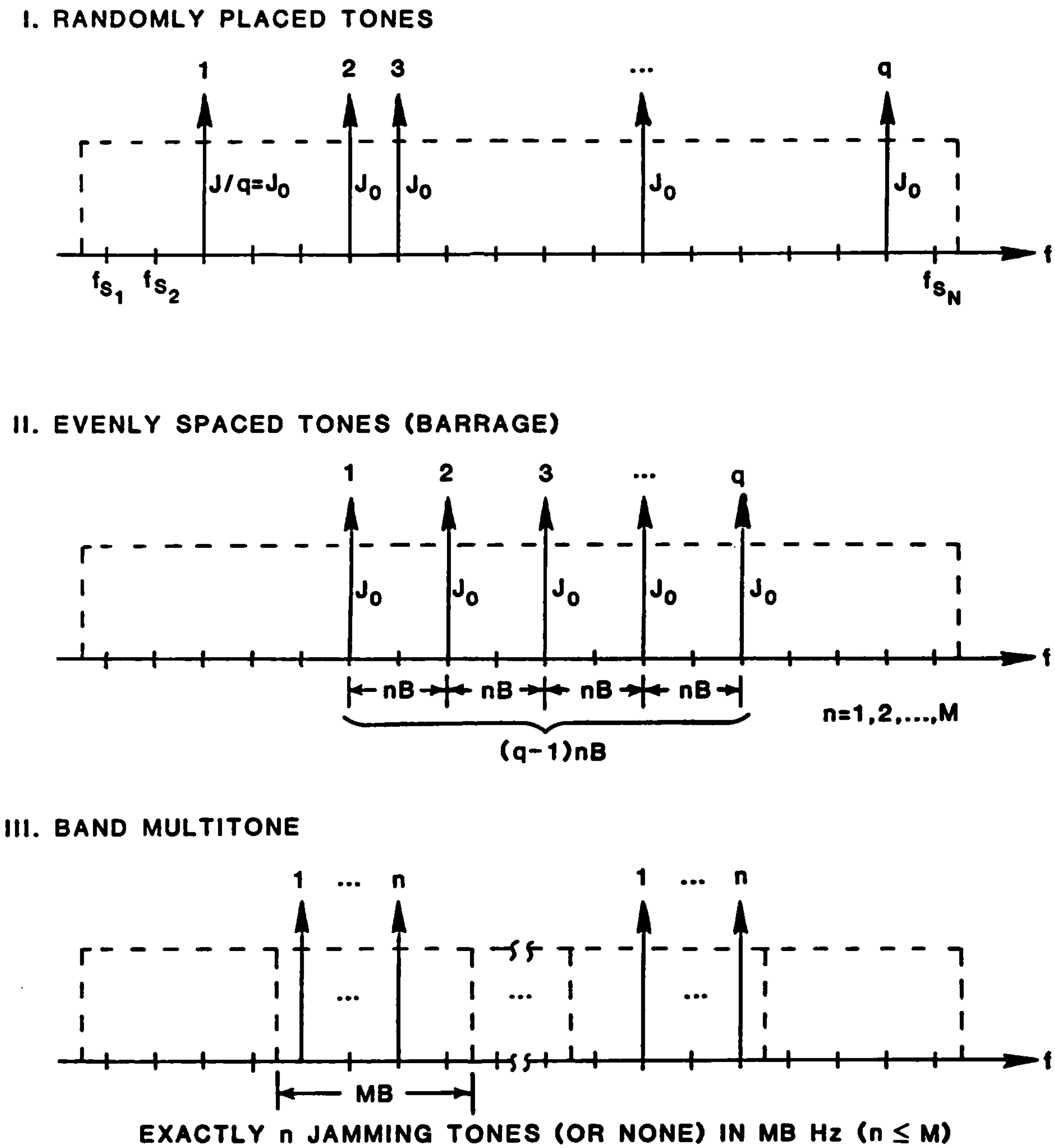


FIGURE 1-12 PARTIAL-BAND TONE JAMMING MODELS

can fall on a jamming tone during a given hop (the cluster of width MB Hz can not span two or more jamming tones separated by nB Hz when $n \geq M$). Thus for this model we restrict the parameters to the range

$$1 \leq q \leq \frac{N}{n} + 1 \tag{1-1a}$$

and

$$1 \leq n \leq M. \tag{1-1b}$$

The third jamming model is known as band multitone jamming [13]. In this model the jamming tones are distributed in such a way that, when jammed, exactly n tones are present in an M-ary cluster. However, the specific filters within the cluster are randomly selected. This model is most appropriate when the hopping takes place in increments of MB Hz rather than B Hz, for it is difficult to conceive of any other way a jammer could insure the exact number of tones (other than n=1) in the cluster when it is jammed (except perhaps for a partial-time follower jammer which, after dehopping, could appear to the system as a band multitone jammer). For band multitone jamming, we restrict the parameters to the range

$$n \leq q \leq \frac{Mn}{N} \tag{1-2a}$$

and

$$1 \leq n \leq M. \tag{1-2b}$$

We emphasize that the definition of the parameter n for band multitone jamming is quite different from the definition of n for barrage jamming.

The analysis of receiver performance in the partial-band tone-jamming channel is more complicated than is the case for the partial-band noise-jamming channel. Therefore, we confine our attention to only one receiver structure, the square-law linear combining receiver. We chose to analyze this receiver in the partial-band tone-jamming channel because the linear combining posed

the fewest analytical obstacles. Even with this choice, when we included thermal noise effects in addition to the jamming tones, it became necessary to resort to approximations to the density function of the output of the signal channel to obtain numerically useful formulations for efficient computations. These approximations, which are discussed more fully in Section 8 of this report, give reasonably good agreement with the exact formulation as shown in Figure 1-13.

An example of the results for the performance of the square-law linear combining receiver in the partial-band tone jamming channel is given in Figure 1-14 for randomly placed jamming tones with the number of jamming tones as a parameter. The figure also shows the envelope of these curves as the performance when the jammer optimizes his choice of the number of jamming tones. We note that the curve of $P_b(e; \gamma_{opt})$ vs. E_b/N_0 follows the curve for $\gamma = 1/N$ for sufficiently high E_b/N_0. This is due to the restriction that $q \geq 1$, and hence $\gamma = q/N \geq 1/N$, which arises from the requirement that the number of tones be a positive integer.

By constructing curves of $P_b(e; \gamma_{opt})$ vs. E_b/N_0 similar to Figure 1-14 for the other jamming models, we are able to produce curves such as that shown in Figure 1-15 to compare various jamming strategies for $\gamma = \gamma_{opt}$. In Figure 1-15 we have plotted the performance of an FH/BFSK receiver with one hop/bit in the various partial-band tone-jamming channels considered, along with the performance in the partial-band noise-jamming channel for comparison. From the figure we see that band multitone jamming with one tone per band and barrage jamming with spacing equal to M are the most effective jamming strategies. For moderate values of E_b/N_J, randomly placed tones are also equally effective, but for E_b/N_J below about 12 dB, the effectiveness of the randomly placed tones falls off. For E_b/N_J above about 35 dB, the curve shows partial-band noise

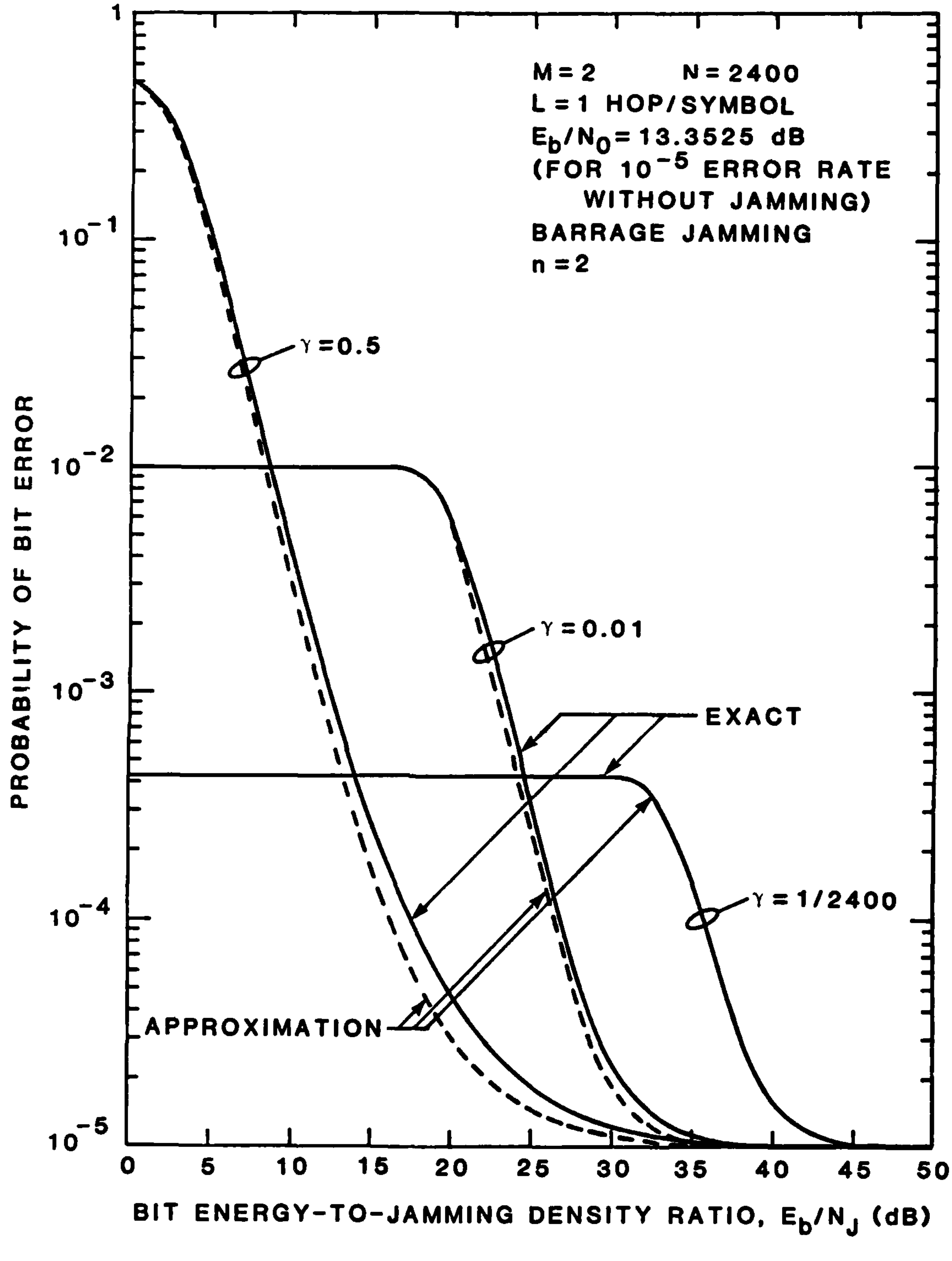


FIGURE 1-13 A COMPARISON OF APPROXIMATE AND EXACT RESULTS FOR TONE JAMMING

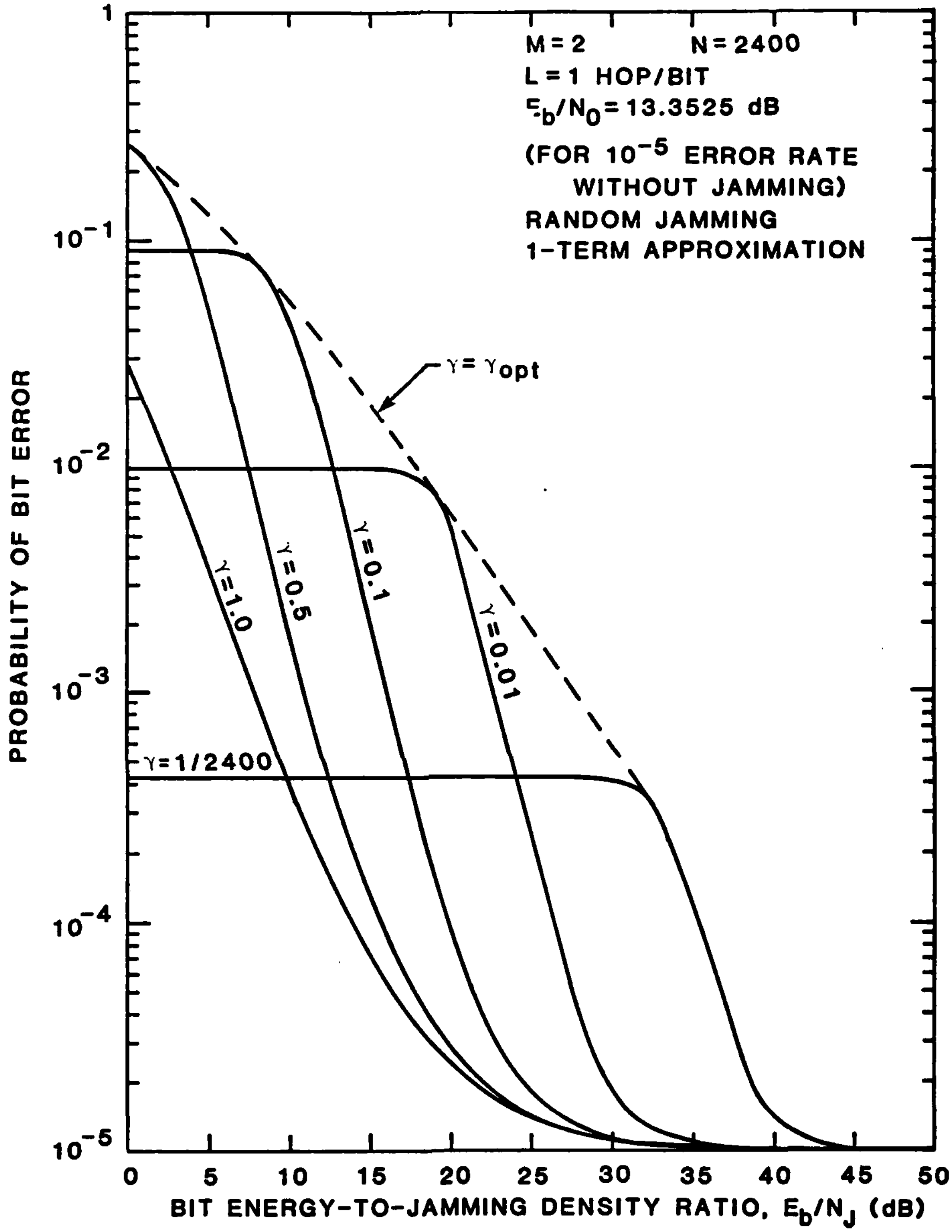


FIGURE 1-14 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR INDEPENDENT MULTITONE JAMMING WHEN L = 1 HOP/BIT AND N = 2400 HOPPING SLOTS FOR BFSK/FH WITH E_b/N_0 = 13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

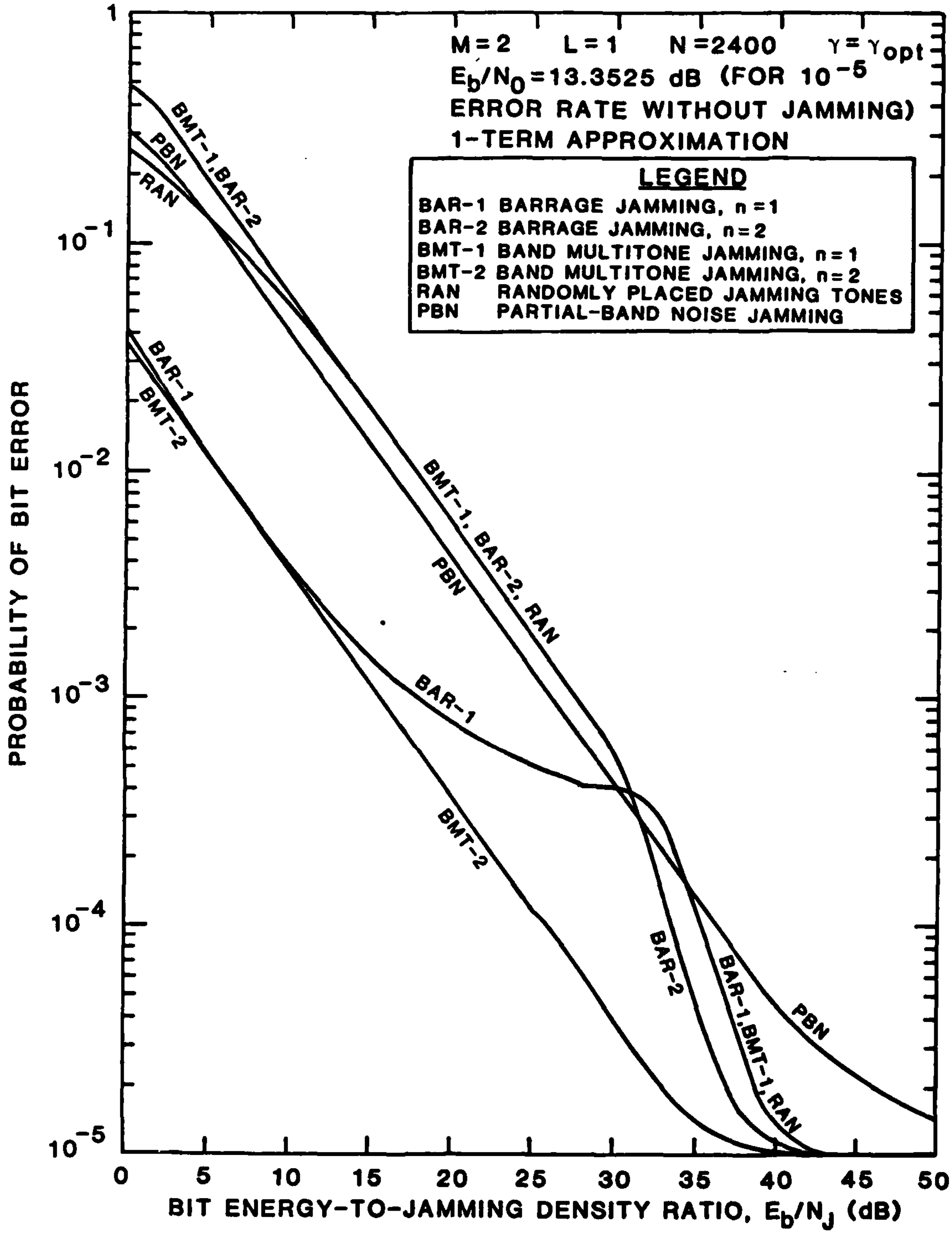


FIGURE 1-15 COMPARISON OF JAMMING STRATEGIES AGAINST SQUARE-LAW LINEAR COMBINING RECEIVER FOR BFSK/FH WITH $L=1$ HOP/BIT, $N=2400$ HOPPING SLOTS, AND $E_b/N_0 = 13.3525$ dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

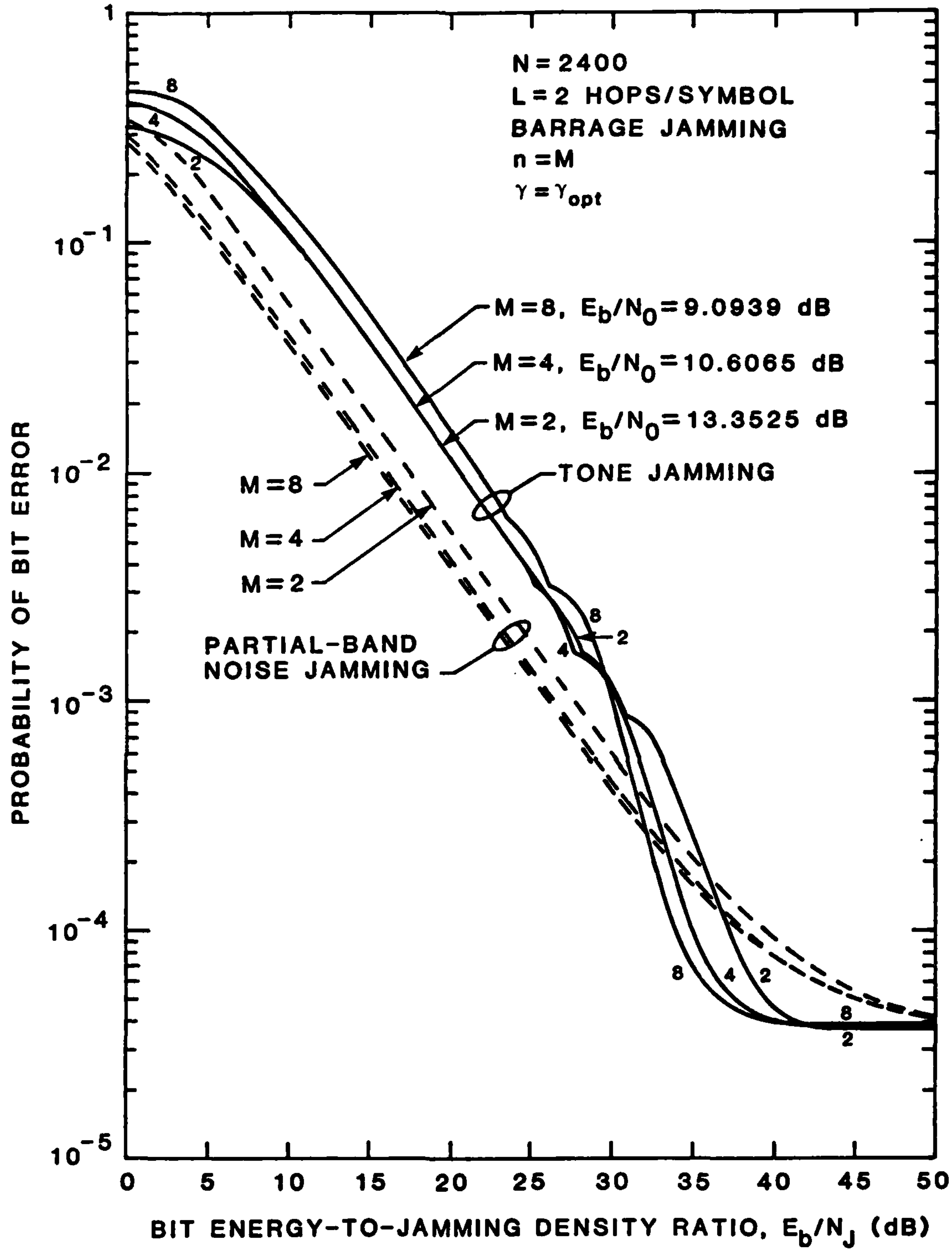


FIGURE 1-16 COMPARISON OF OPTIMUM TONE JAMMING (BARRAGE, n = M) AND OPTIMUM PARTIAL-BAND NOISE JAMMING AGAINST MFSK/FH SQUARE-LAW LINEAR COMBINING RECEIVER WHEN L = 2 HOPS/SYMBOL AND N = 2400 HOPPING SLOTS

jamming to be most effective. This is a result of a slightly different maximization technique used for the partial-band noise-jamming model, wherein the fraction γ was allowed to approach arbitrarily close 0. Were we to limit γ to a value greater than M/N for partial-band noise jamming,* its apparent superiority at high E_b/N_J would not remain.

The comparison of the several tone-jamming models was performed for M = 2 using an approximation in the density function of the output of the signal channel to obtain numerical results with a reasonable expenditure of computer time. Once we identified the worst-case tone-jamming scenario as barrage jamming with spacing parameter n = M, we reverted to the exact analysis to examine the performance of systems with higher values of M under this worst-case scenario. The results obtained from this analysis using the exact equations are typified by Figure 1-16 which shows the performance of a system with L = 2 hops per symbol under worst-case barrage tone jamming with tone-spacing parameter n = M, where M is a parameter.

We see from Figure 1-16 rather startling behavior of the performance under tone jamming: as M increases from 4 to 8, the performance degrades. Careful examination of the physical situation provides the explanation. When M is increased, the jamming power per jamming tone, under optimum choice of the number of tones, and for a fixed E_b/N_J, is higher relative to the signal power for M = 8 than for M = 4. This is possible because the increased frequency occupancy of the 8-ary signal cluster (8B vs. 4B for the 4-ary system) allows the jammer to use fewer tones while maintaining a high probability of causing interference. This situation is discussed in greater detail in Section 8.3.3.

---

*For partial-band noise jamming the minimum γ is M/N because of our assumption that the entire M-ary cluster is either jammed or unjammed.

1.3.3 Concluding Remarks

We have analyzed the performance of FH/MFSK receivers for an L-hops/symbol transmission scheme in the partial-band noise-jamming and partial-band tone-jamming channels. Our analyses have taken the important step, which was missing in prior work, of including the effects of thermal noise. Our numerical results show that, under strong jamming conditions, a limited M-ary coding gain is achieved. For the nonlinear combining receivers, a very limited amount of quasi-diversity improvement may be gained by increasing the number of hops per symbol, but only over a limited range of E_b/N_J.

These results are new, and demonstrate that neglecting thermal noise in the analysis can produce misleading results. Further, it shows that bounding techniques, such as the union bound, are not always appropriate for the partial-band jamming channels.

Finally, our results also show that use of multiple hops per symbol is not uniformly effective in countering the effects of partial-band jamming. The use of multiple hops per symbol should be viewed not as an anti-jam measure, but as solely a low-probability-of-intercept measure. A slight AJ gain can be achieved by use of higher-order M-ary alphabets in the partial-band noise-jamming channel, but this should be used with caution because it may actually degrade performance in the partial-band tone-jamming channel if the jammer is able to optimize his tone spacing and number of tones for the specific modulation used by his victim.

2.0 PERFORMANCE OF CONVENTIONAL SQUARE-LAW LINEAR COMBINING RECEIVER

The first receiver which we analyze is the conventional square-law envelope detector with linear combining of the multiple hops per M-ary symbol. This is a well-known receiver structure, based on the maximum likelihood structure for the Gaussian channel. In the following discussion, we obtain expressions for the performance of the square-law linear combining receiver in the Gaussian (thermal) noise channel, and under both wideband and partial-band noise jamming conditions. The section concludes with a selection of numerical results presented in graphical and tabular form.

The square-law linear combining receiver is presented as a baseline for comparison of other more complicated receiver structures. Although the square-law linear combining receiver is a reasonably close approximation to the optimum receiver for the Gaussian channel, we would expect that some other structure may provide better performance on a non-Gaussian channel such as the partial-band jamming channel. Indeed, we show in subsequent chapters of this report that this is the case.

2.1 PERFORMANCE OF SQUARE-LAW LINEAR COMBINING RECEIVER IN THE GAUSSIAN NOISE CHANNEL

The M-ary FSK modulation conveys information by transmitting one of $M=2^K$ symbols every T_s seconds, where the symbol to be transmitted is determined by a group of K data bits. If the information source produces bits at a rate $R_b = 1/T_b$, then one M-ary symbol must be transmitted every $T_s = KT_b$ seconds. The M-ary symbol is conveyed to the receiver by selecting one of M frequencies to be transmitted. We assume the frequencies are spaced evenly across a contiguous band with the spacing chosen as the reciprocal of the pulse width to obtain orthogonal signals.

In order to provide some degree of protection against jamming, the signal is further subjected to a spread spectrum modulation in the form of frequency hopping. Every $\tau = T_s/L$ seconds the location of the M-ary cluster of M frequencies is hopped over a wide band W, thereby dividing the M-ary symbol of duration T_s into L hops each of duration τ. Thus the bandwidth of the pulse transmitted on any one hop is $B \triangleq 1/\tau$; and the bandwidth of the M-ary cluster will be $MB = M/\tau$. If the total system bandwidth is W, then $N = W/B - M+1$ hopping locations are available; if $W/B >> M$ we may make the approximation $N \approx W/B$.

The receiver for the MFSK/FH signal is shown in Figure 2-1. The incoming signal is dehopped by mixing with a synchronized replica of the hopping oscillator at the transmitter. The dehopped signal (plus noise and jamming) is then applied to a bank of M bandpass filters, each of width B, centered at the M possible signalling frequencies. The output of each filter is processed by a square-law envelope detector (i.e. a device whose output voltage is proportional to the square of the envelope of the input signal). Each squared envelope is sampled once every τ seconds. The L samples from the L hops in one symbol are summed for each channel of the receiver. At the end of L hops the sums are compared, the largest sum is selected, and the symbol decision is made on the basis of which channel has this largest sum.

2.1.1 Performance Analysis

The dehopped received waveform r(t) may be represented during any given hop as

$$r(t) = s(t) + n(t) \tag{2-1}$$

where s(t) is the information-bearing signal and n(t) is bandlimited white Gaussian thermal noise. Over a hop interval the signal s(t) at the output of the dehopping mixer in the receiver is

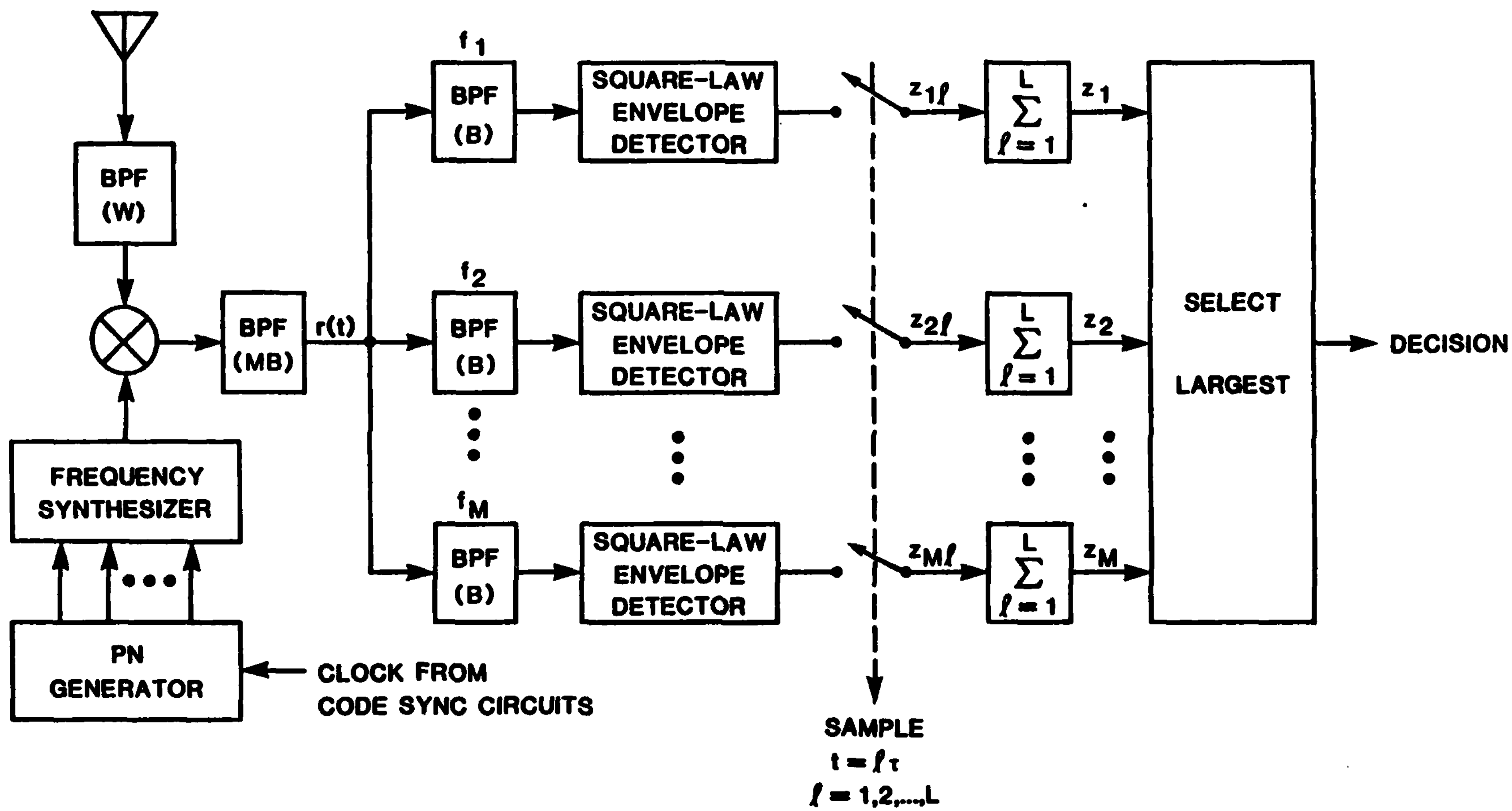


FIGURE 2-1 RECEIVER FOR MFSK/FH WITH FAST HOPPING

$$s(t) = \sqrt{2S}\cos(2\pi f_i t + \theta_i), \text{ symbol "i" transmitted,}$$

$$i = 1, 2, \ldots, M, \tag{2-2}$$

where S is the received (average) signal power; f_i is the frequency for the i-th symbol, $i = 1, 2, \ldots, M$; and θ_i, $i = 1, 2, \ldots, M$ are independent phases uniformly distributed on $[0, 2\pi)$. The thermal noise n(t) may be expressed in the form of a Rician decomposition,

$$n_i(t) = n_{ci}(t)\cos 2\pi f_i t + n_{si}(t)\sin 2\pi f_i t; \; i = 1, 2, \ldots, M, \tag{2-3}$$

where $n_{ci}(t)$ and $n_{si}(t)$ at a given time are statistically independent Gaussian random variables with variances (or average power) given by

$$E\left[n_i^2(t)\right] = E\left[n_{ci}^2(t)\right] = E\left[n_{si}^2(t)\right] = \sigma_N^2 = N_0 B \tag{2-4}$$

where N_0 is the noise density in watts per hertz.

Without loss of generality, we may assume that the symbol "1" is transmitted. Then each of the squared envelope samples $z_{1\ell}$ is a scaled non-central chi-squared (χ^2) variate with two degees of freedom. The density function is

$$p_{z_{1\ell}}(\alpha) = \frac{1}{2\sigma_N^2}\exp\left(-\frac{\alpha}{2\sigma_N^2} - \rho_N\right) I_0\left(\sqrt{\frac{2\alpha\rho_N}{\sigma_N^2}}\right) \tag{2-5}$$

where

$$\rho_N \triangleq \frac{S}{\sigma_N^2} \tag{2-6a}$$

is the signal-to-noise ratio for one hop. If the symbol energy is E_s, then

$$\rho_N = \frac{E_s}{LN_0}, \tag{2-6b}$$

which may also be written as

$$\rho_N = \frac{KE_b}{LN_0} \tag{2-6c}$$

where E_b/N_0 is the bit energy-to-noise density ratio.

The squared envelope samples of the noise-only channels, $z_{i\ell}$, $i = 2, 3, \ldots, M$, are each scaled central χ^2 variates with two degrees of freedom and density function

$$p_{z_{1\ell}}(\alpha) = \frac{1}{2\sigma_N^2} \exp\left(-\frac{\alpha}{2\sigma^2}\right), \quad i = 2, 3, \ldots, M. \tag{2-7}$$

Since the sum of χ^2 variates is another χ^2 variate, the density of the sums of samples taken over one symbol, as shown in Figure 2-1, is given by

$$p_{z_1}(\alpha) = \frac{1}{2\sigma_N^2}\left(\frac{\alpha}{2\sigma^2\rho_N}\right)^{(L-1)/2} \exp\left(-\frac{\alpha}{2\sigma_N^2} - L\rho_N\right) I_{L-1}\left(\sqrt{\frac{2L\alpha\rho_N}{\sigma_N^2}}\right) \tag{2-8}$$

for the signal channel and

$$p_{z_i}(\beta) = \frac{1}{2\sigma_N^2}\left(\frac{\beta}{2\sigma_N^2}\right)^{L-1} \frac{1}{\Gamma(L)} \exp\left(-\frac{\beta}{2\sigma_N^2}\right), \quad i = 2, 3, \ldots, M \tag{2-9}$$

for the noise-only channels. The probability of making an incorrect symbol decision is

$$P_s(e) = \text{Prob}\left\{z_1 < z_2 \text{ or } z_1 < z_3 \text{ or } \ldots \text{ or } z_1 < z_M\right\} \tag{2-10}$$

or equivalently

$$P_s(e) = 1 - \text{Prob}\left\{z_1 > z_2 \text{ and } z_1 > z_3 \text{ and } \ldots \text{ and } z_1 > z_M\right\}. \tag{2-11}$$

In terms of the density functions, (2-11) can be written as

$$P_s(e) = 1 - \int_0^\infty p_{z_1}(\alpha)\left[\int_0^\alpha p_{z_2}(\beta)d\beta\right]^{M-1} d\alpha. \tag{2-12a}$$

Since the density $p_{z_1}(\alpha)$ integrates to 1, we may also write (2-12a) in the form

$$P_s(e) = \int_0^\infty p_{z_1}(\alpha)\left\{1 - \left[\int_0^\alpha p_{z_2}(\beta)d\beta\right]^{M-1}\right\}d\alpha. \tag{2-12b}$$

Substitution of (2-8) and (2-9) into (2-12b) yields

$$P_s(e) = e^{-L\rho_N}\int_0^\infty \frac{1}{2\sigma_N^2}\left(\frac{\alpha}{2\sigma_N^2\rho_N}\right)^{(L-1)/2} \exp\left(-\frac{\alpha}{2\sigma_N^2}\right) I_{L-1}\left(\sqrt{\frac{2L\alpha\rho_N}{\sigma_N^2}}\right)$$

$$\cdot \left\{1 - \left[\int_0^\alpha \frac{1}{2\sigma_N^2}\left(\frac{\beta}{2\sigma_N^2}\right)^{L-1} \frac{1}{\Gamma(L)} \exp\left(-\frac{\beta}{2\sigma_N^2}\right)d\beta\right]^{M-1}\right\} d\alpha. \tag{2-13}$$

Upon making the changes of variables $x = \alpha/2\sigma_N^2$ and $y = \beta/2\sigma_N^2$ in (2-13), we obtain

$$P_s(e) = e^{-L\rho_N}\int_0^\infty \left(\frac{x}{\rho_N}\right)^{(L-1)/2} e^{-x}\, I_{L-1}\left(\sqrt{4L\rho_N x}\right)\left\{1 - \left[\int_0^x y^{L-1} \frac{1}{\Gamma(L)} e^{-y}dy\right]^{M-1}\right\} dx. \tag{2-14}$$

The inner integral in (2-14) may be evaluated using [2, eq. 3.381.1] to yield the form

$$P_s(e) = e^{-L\rho_N}\int_0^\infty \left(\frac{x}{\rho_N}\right)^{(L-1)/2} e^{-x}\, I_{L-1}\left(\sqrt{4L\rho_N x}\right)$$

$$\cdot \left\{1 - \left[\frac{\gamma(L,x)}{\Gamma(L)}\right]^{M-1}\right\}dx \tag{2-15}$$

where $\gamma(L,x)$ is an incomplete gamma function which may be represented by the finite series [2, eq. 8.352.1]

$$\gamma(L,x) = \Gamma(L) \left[1 - e^{-x} \sum_{k=0}^{L-1} \frac{x^k}{k!}\right]. \tag{2-16}$$

Substitution of (2-16) into (2-15) and the binomial theorem yield the form

$$P_s(e) = e^{-L\rho_N} \int_0^{\infty} \left(\frac{x}{\rho_N}\right)^{(L-1)/2} e^{-x} I_{L-1}\left(\sqrt{4L\rho_N x}\right) \left\{1 - \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m \left[\sum_{k=0}^{L-1} \frac{x^k}{k!}\right]^m e^{-mx}\right\} dx. \tag{2-17}$$

Recognizing that the first term of the summation over m in (2-17) is equal to 1, we obtain

$$P_s(e) = e^{-L\rho_N} \int_0^{\infty} \left(\frac{x}{\rho_N}\right)^{(L-1)/2} e^{-x} I_{L-1}\left(\sqrt{4L\rho_N x}\right) \sum_{m=1}^{M-1} \binom{M-1}{m} (-1)^{m+1} \left[\sum_{k=0}^{L-1} \frac{x^k}{k!}\right]^m e^{-mx} \, dx. \tag{2-18}$$

To evaluate the power of a summation in (2-18), we use the J.C.P. Miller Formula [3, p. 42],

$$\left[\sum_{i=0}^{\infty} b_i x^i\right]^m = \sum_{j=0}^{\infty} a_{j,m} x^j , \quad b_0 = 1, \tag{2-19}$$

where the coefficients $a_{j,m}$ are defined by the recurrence relation

$$a_{j,m} = \frac{1}{j} \sum_{q=1}^{j} \left[(m+1)q - j\right] a_{j-q,m} \, b_q \tag{2-20a}$$

with

$$a_{0,m} = 1. \tag{2-20b}$$

If we define $b'_q \triangleq b_q/q!$, then (2-20) becomes

$$c_{j,m} = \frac{1}{j}\sum_{q=1}^{j}\binom{j}{q}[(m+1)q-j]\; c_{j-q,m}\, b'_q \qquad (2\text{-}21a)$$

$$c_{j,0} = 1 \qquad (2\text{-}21b)$$

where $c_{j,m} = j!a_{j,m}$. Using (2-21) with

$$b'_q = \begin{cases} 1, & q \leq L-1 \\ 0, & q > L-1 \end{cases} \qquad (2\text{-}22)$$

in (2-18), and interchanging the order of integration and summation, we have

$$P_s(e) = e^{-L\rho_N}\sum_{m=1}^{M-1}\binom{M-1}{m}(-1)^{m+1}\sum_{j=0}^{m(L-1)}\frac{c_{j,m}}{j!}\left(\frac{1}{\rho_N}\right)^{(L-1)/2}\int_0^{\infty} x^{(L-1)/2+j}\; e^{-(m+1)x}$$

$$\cdot\; I_{L-1}\left(\sqrt{4L\rho_N x}\right) dx \qquad (2\text{-}23a)$$

where

$$c_{j,m} = \frac{1}{j}\sum_{q=1}^{j}\binom{j}{q}[(m+1)q-j]\; c_{j-q,m}\, b'_q\,, \quad 1 \leq j \leq m(L-1) \qquad (2\text{-}23b)$$

$$c_{0,m} = 1 \qquad (2\text{-}23c)$$

with b'_q as given by (2-22), and we recognize that the expansion of the power of the summation in (2-18) will terminate at $x^{m(L-1)}$.

The integral in (2-23a) can be evaluated in terms of the confluent hypergeometric function (Kummer's function) using [2, eq. 6.643.2 and 9.220.1] to yield

$$P_s(e) = e^{-L\rho_N} \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m+1}}{(m+1)^L} \sum_{j=0}^{m(L-1)} \frac{\Gamma(L+j)}{\Gamma(L)} \frac{c_{j,m}}{j!(m+1)^j} \, {}_1F_1\left(L+j;\, L;\, \frac{L\rho_N}{m+1}\right). \tag{2-24}$$

Using Kummer's transformation [4, eq. 13.1.27] and the relationship between the generalized Laguerre polynomials and the confluent hypergeometric function [4, eq. 13.6.9], we can write (2-24) in the form

$$P_s(e) = \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m+1}}{(m+1)^L} \sum_{j=0}^{m(L-1)} \frac{c_{j,m}}{(m+1)^j} \exp\left(-\frac{mL\rho_N}{m+1}\right) \mathcal{L}_j^{(L-1)}\left(-\frac{L\rho_N}{m+1}\right) \tag{2-25}$$

where $\mathcal{L}_j^{(L-1)}$ is the generalized Laguerre polynomial [4, eq. 22.3.9] which may be computed recursively using the relation [2, eq. 8.971.6]

$$(n+1)\, \mathcal{L}_{n+1}^{(\alpha)}(x) = (2n+\alpha+1-x)\, \mathcal{L}_n^{(\alpha)}(x) - (n+\alpha)\, \mathcal{L}_{n-1}^{(\alpha)}(x). \tag{2-26}$$

The conversion from symbol (or word) error probability to bit error probability for orthogonal signalling is given by [5, p. 198]

$$P_b(e) = \frac{M}{2(M-1)} P_s(e). \tag{2-27}$$

From (2-25) and (2-27), then, we have the final result

$$P_b(e) = \frac{M}{2(M-1)} \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m+1}}{(m+1)^L} \sum_{j=0}^{m(L-1)} \frac{c_{j,m}}{(m+1)^j} \exp\left(-\frac{mL\rho_N}{m+1}\right) \mathcal{L}_j^{(L-1)}\left(-\frac{L\rho_N}{m-1}\right) \tag{2-28}$$

where the coefficientes $c_{j,m}$ are given by (2-24). Typical performance curves computed using (2-28) are shown in Figures 2-2 and 2-3. Figure 2-2 shows bit error probability as a function of the bit energy-to-noise density ratio, E_b/N_0, with the alphabet size (M) as a parameter for a fixed number of hops per symbol. We observe that an "M-ary coding gain" is achieved by

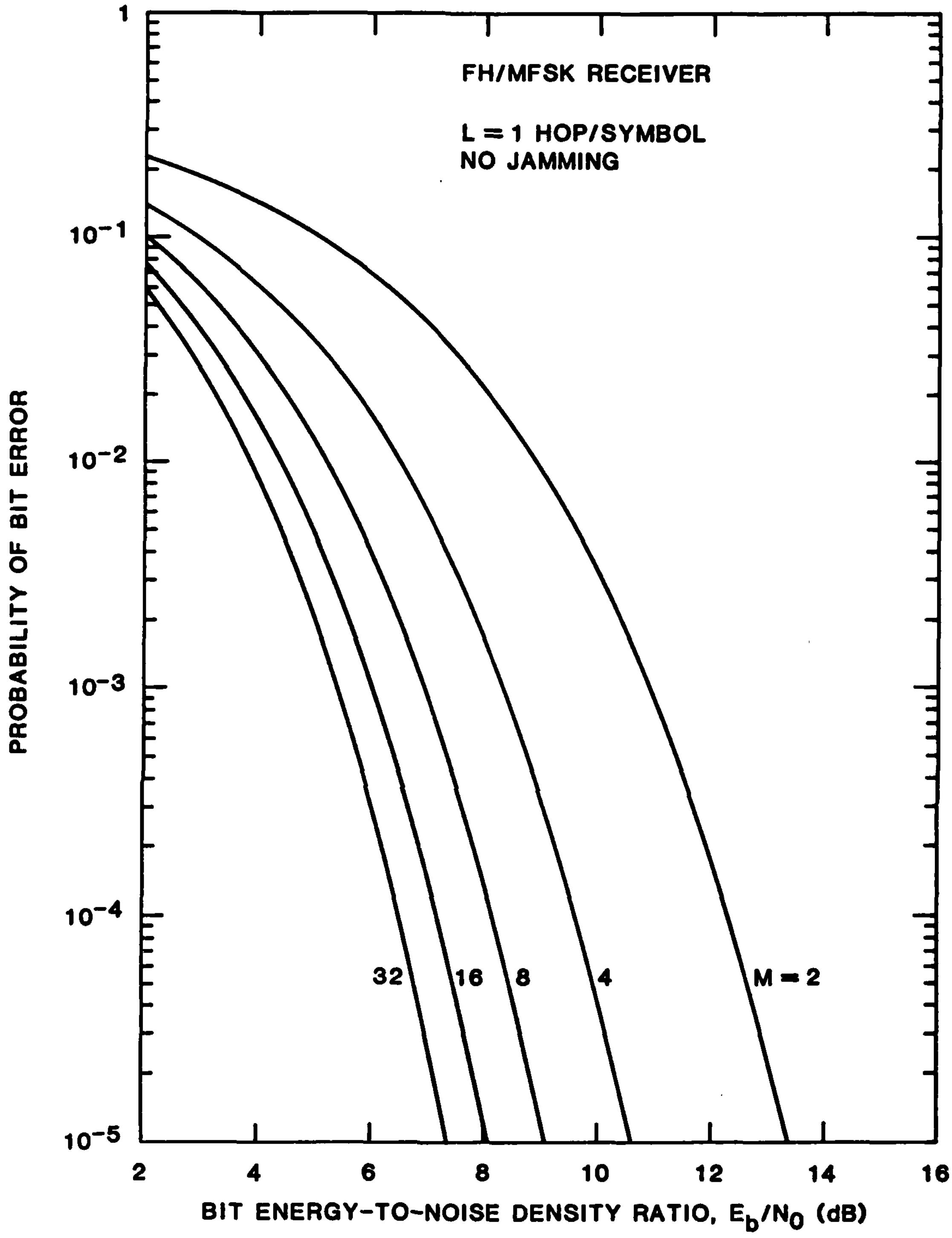


FIGURE 2-2 PROBABILITY OF BIT ERROR VS. BIT ENERGY-TO-NOISE DENSITY RATIO FOR LINEAR COMBINING RECEIVER IN GAUSSIAN CHANNEL (NO JAMMING) FOR L=1 HOP/SYMBOL WITH THE ALPHABET SIZE (M) VARIED

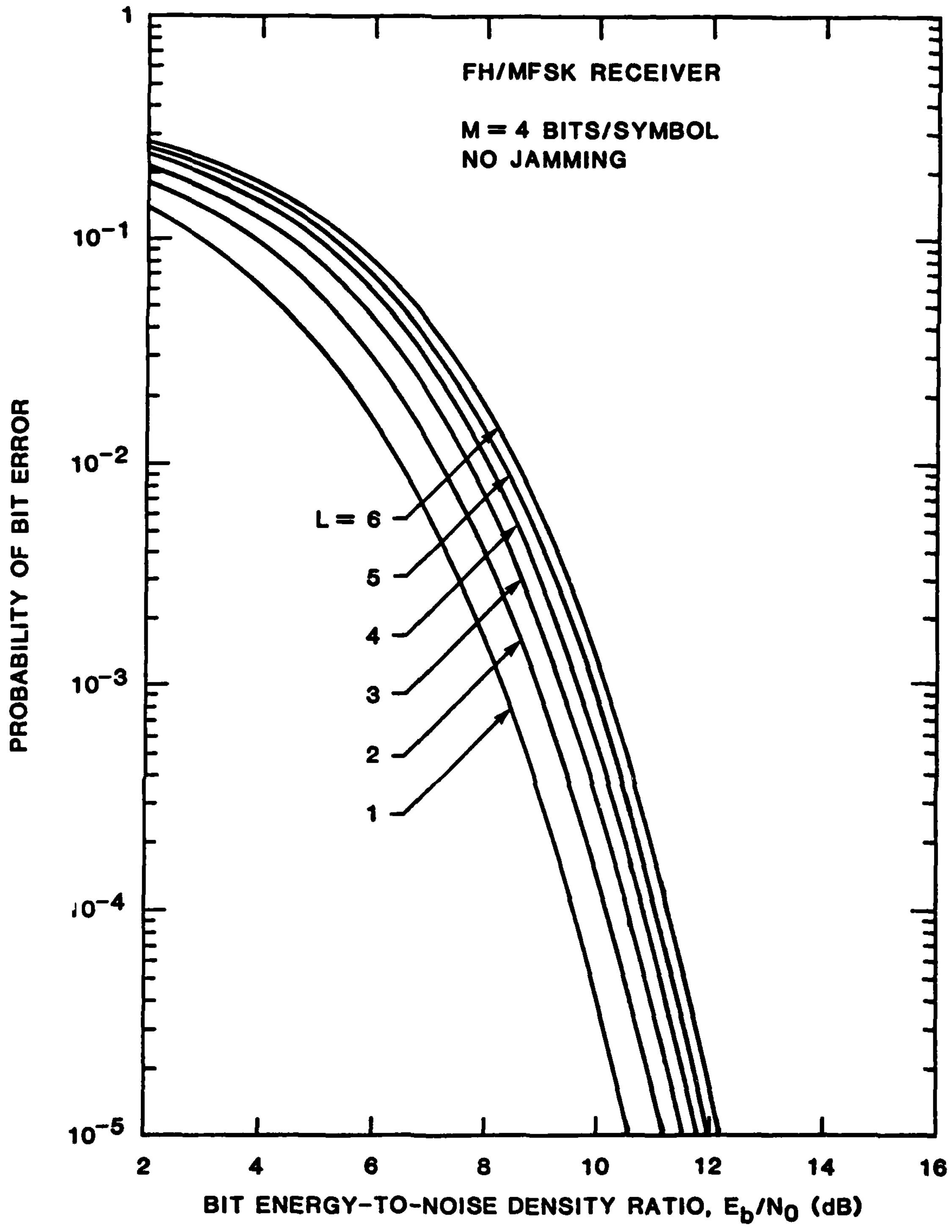


FIGURE 2-3 PROBABILITY OF BIT ERROR VS. BIT ENERGY-TO-NOISE DENSITY RATIO FOR FH/MFSK (M = 4) SQUARE-LAW LINEAR COMBINING RECEIVER IN GAUSSIAN CHANNEL (NO JAMMING) WITH THE NUMBER OF HOPS/SYMBOL (L) VARIED

increasing the alphabet size. Figure 2-3 shows the bit error probability as a function of E_b/N_0 for a constant alphabet size with L as a parameter. We note that increasing L degrades the performance. This phenomenon is discussed below.

2.1.2 Noncoherent Combining Loss

For post-detection combining (noncoherent combining) on a nonfading Gaussian channel, once a symbol is split into a number of pieces ($L > 1$), the original performance ($L=1$) can never be achieved. As shown in Figure 2-4, for $L > 1$ hop/symbol the bit energy-to-noise density ratio required to achieve a specified bit error probability must be increased by an amount d_L beyond that required to achieve the same bit error probability with L=1. The quantity d_L (in dB) is termed the noncoherent combining loss. In general, the noncoherent combining loss will be a function of both L *and the value of* $P_b(e)$ at which it is measured.

It is very difficult to obtain a useful analytical formula for d_L. To appreciate the difficulties involved, consider that $P_b(e)$ is a function of ρ_N, L, and M as given by (2-28). For brevity, let us write

$$P_b(e) = f(\rho_N;\, L,\, M) \tag{2-29}$$

and define the inverse function with respect to the first argument

$$\rho_N = f^{-1}(P_b;\, L,\, M) \tag{2-30}$$

where (2-30) gives an answer for ρ_N as a numeric ratio (not dB). Then the noncoherent combining loss at a specified P_b, say P_0, is given (in decibels) by

$$d_L = 10 \log_{10} \left[\frac{f^{-1}(P_0;\, L,\, M)}{f^{-1}(P_0;\, 1,\, M)} \right] . \tag{2-31}$$

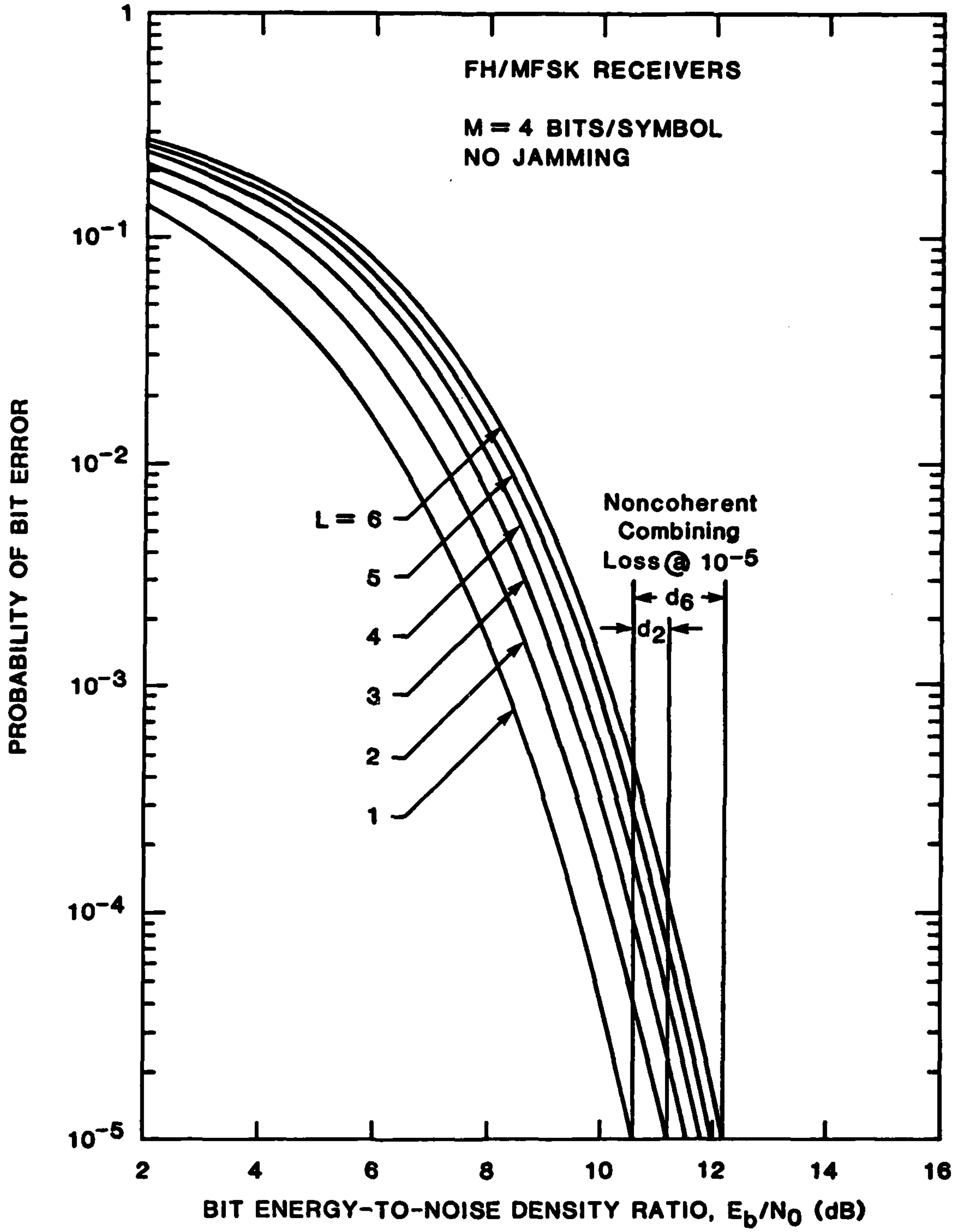


FIGURE 2-4 DEFINITION OF NONCOHERENT COMBINING LOSS

Looking back at (2-28), we see that determining the inverse function $f^{-1}(P_b; L, M)$ involves inverting a product of an exponential and a polynomial of degree (M-1)(L-1) for which the coefficients are known only through a recursion relation. Indeed, solving (2-30) explicitly is a very formidable task. It is much more convenient to determine combining loss by numerical methods.

Figures 2-5 and 2-6 show the noncoherent combining loss as a function of the number of hops per symbol with M as a parameter for $P_b(e) = 10^{-3}$ and 10^{-5}, respectively. We observe that the noncoherent combining loss is lower for lower values of $P_b(e)$ (Figure 2-5 vs. Figure 2-6) and also decreases as M, the alphabet size, increases.

The observation that noncoherent combining loss decreases as M increases, while true, is somewhat misleading. Let us define a new term "noncoherent combining penalty", δ_L, which measures the increase in bit error rate as L increases while E_b/N_0 is held constant. We can express this quantity as

$$\delta_L = \frac{f(\rho_N;\ L,\ M)}{f(\rho_N;\ 1,\ M)}\ . \tag{2-32}$$

Equation (2-32) also has an advantage over (2-31) in that it does not require finding $f^{-1}(P_b)$. We have plotted this noncoherent combining penalty in Figures 2-7 and 2-8 for $P_b(e) = 10^{-3}$ and 10^{-5}, respectively. In examining these figures, remember that large δ_L is bad: the bit error rate is δ_L times higher than the bit error rate for L=1. Note that the trend with increasing M in Figures 2-7 and 2-8 is opposite from that in Figures 2-5 and 2-6; as M increases, so does δ_L. The reason for this is that the slope of the curves of $P_b(e)$ vs. E_b/N_0 increases as M increases; hence even though the change δ_L at a fixed E_b/N_0 increases as M increases, a smaller increment d_L is needed to bring $P_b(e)$ down to the reference value.

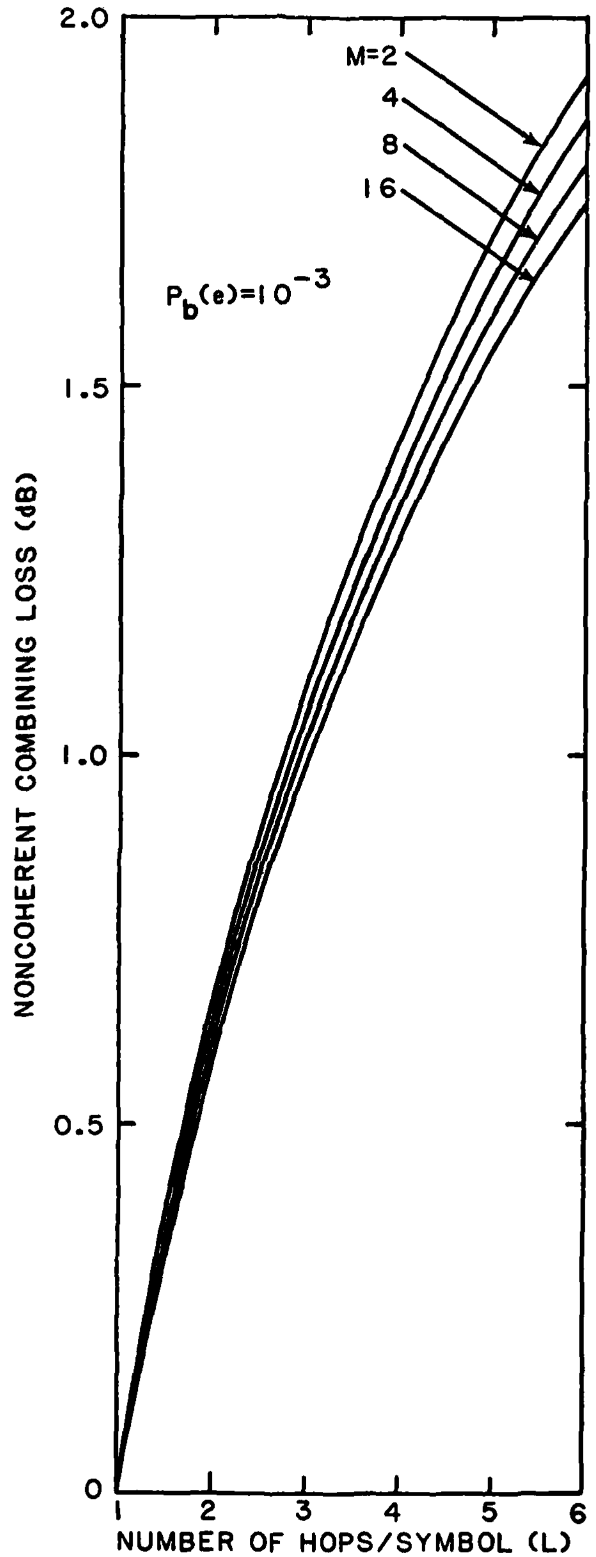


FIGURE 2-5 NONCOHERENT COMBINING LOSS AS A FUNCTION OF NUMBER OF HOPS/SYMBOL AT $P_b(e)=10^{-3}$ FOR SQUARE-LAW LINEAR COMBINING RECEIVER WITH THE ALPHABET SIZE (M) VARIED

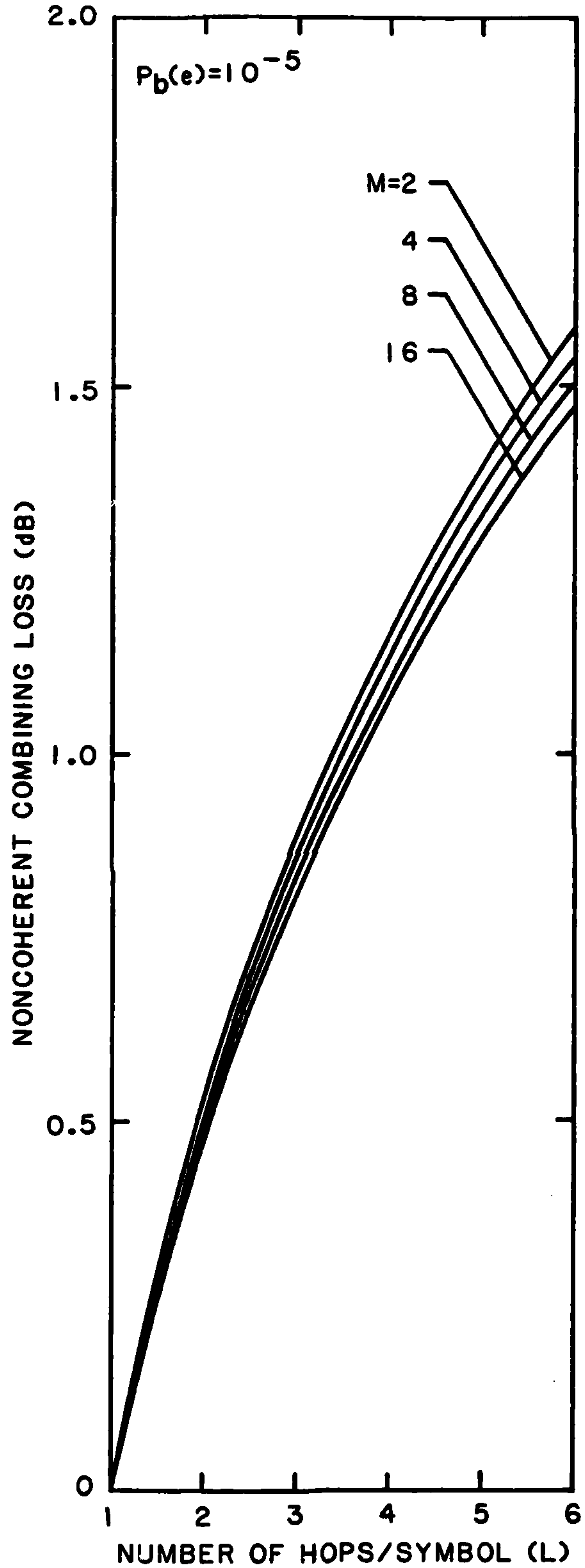


FIGURE 2-6 NONCOHERENT COMBINING LOSS AS A FUNCTION OF NUMBER OF HOPS/SYMBOL AT $P_b(e)=10^{-5}$ FOR SQUARE-LAW LINEAR COMBINING RECEIVER WITH THE ALPHABET SIZE (M) VARIED

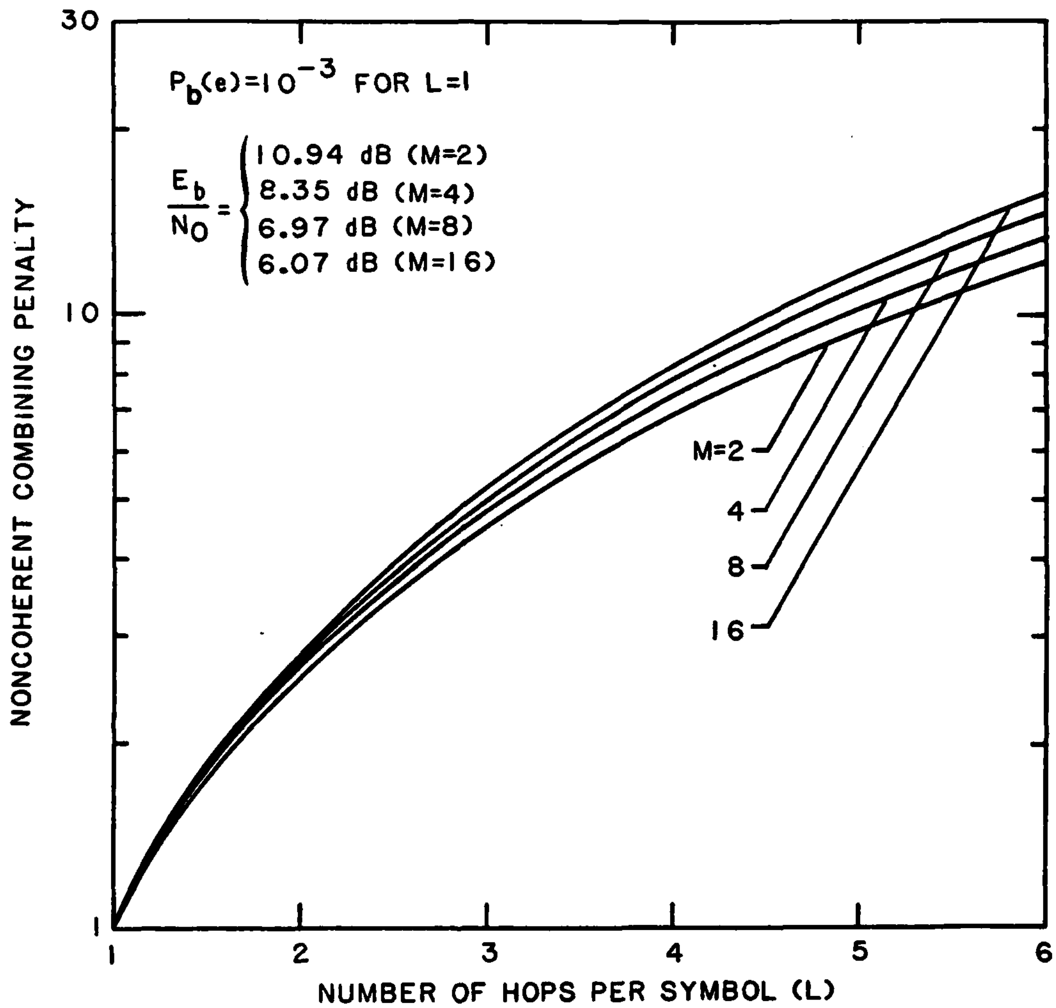


FIGURE 2-7 NONCOHERENT COMBINING PENALTY AS A FUNCTION OF NUMBER OF HOPS/SYMBOL AT E_b/N_0 CORRESPONDING TO $P_b(e)=10^{-3}$ FOR L=1 HOP/SYMBOL WITH THE ALPHABET SIZE (M) VARIED

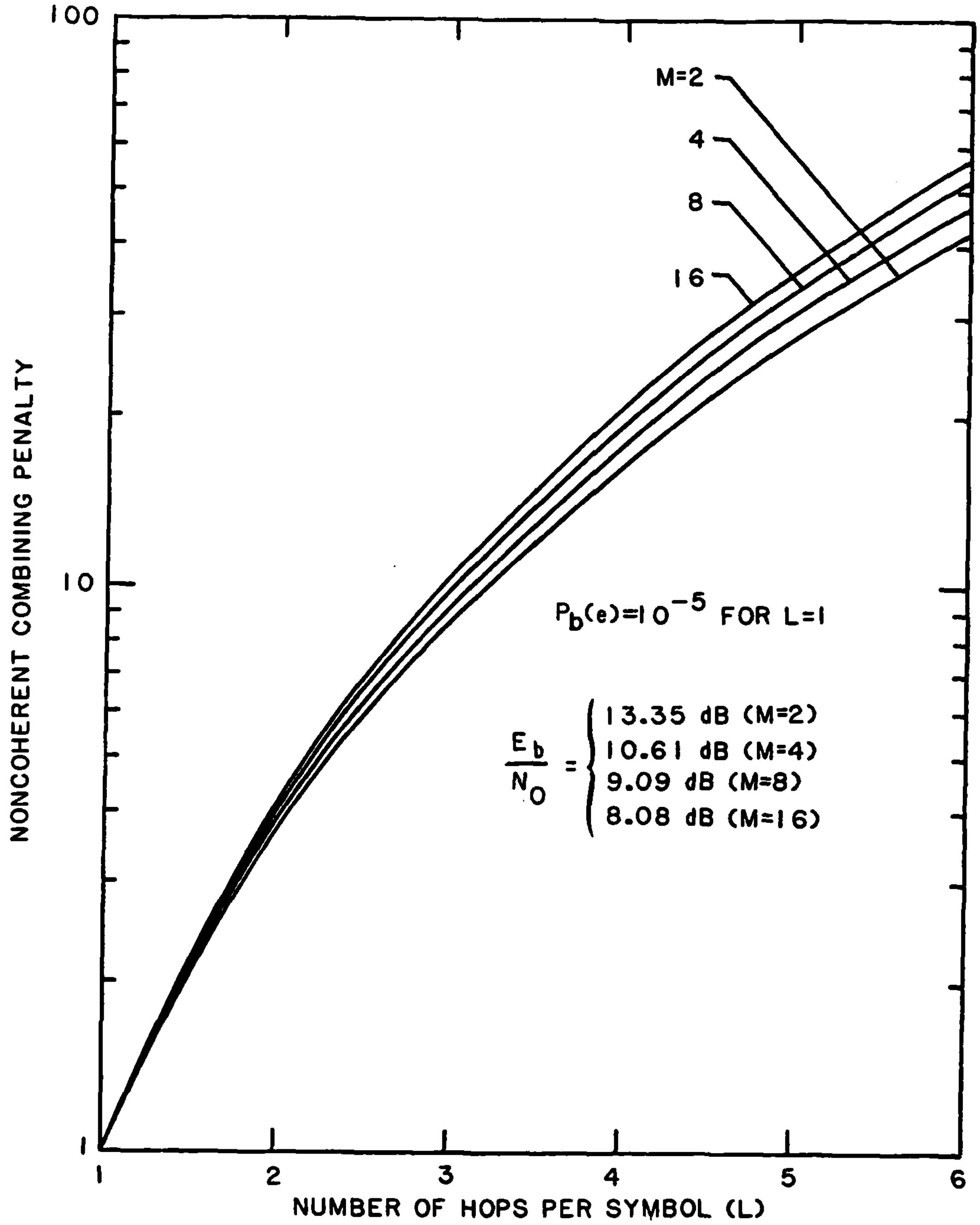


FIGURE 2-8 NONCOHERENT COMBINING PENALTY AS A FUNCTION OF NUMBER OF HOPS/SYMBOL AT E_b/N_0 CORRESPONDING TO $P_b(e)=10^{-5}$ FOR L=1 HOP/SYMBOL WITH THE ALPHABET SIZE (M) VARIED

2.2 PERFORMANCE OF SQUARE-LAW LINEAR COMBINING RECEIVER IN THE WIDEBAND JAMMING CHANNEL

The wideband jamming channel is characterized by the presence of Gaussian jamming noise of bandwidth W with total jamming power J. Hence the jamming noise density is

$$N_J = J/W. \tag{2-33}$$

The jamming noise in each receiver filter may be represented by a Rician decomposition

$$j_i(t) = j_{ci}(t) \cos 2\pi f_i t + j_{si}(t) \sin 2\pi f_i t,$$
$$i = 1, 2, \ldots, M, \tag{2-34}$$

where $j_{ci}(t)$ and $j_{si}(t)$ at a given time are statistically independent zero-mean Gaussian random variables with variances (or average power) given by

$$\sigma_J^2 \triangleq E\left\{j_i^2(t)\right\} = E\left\{j_{ci}^2(t)\right\} = E\left\{j_{si}^2(t)\right\} = N_J. \tag{2-35}$$

The jamming noise is also statistically independent of the thermal noise.

Since the two noises are additive, the total noise power at the output of a receiver filter is

$$\sigma_T^2 \triangleq \sigma_N^2 + \sigma_J^2. \tag{2-36}$$

Therefore, the results of Section 2.1 may be applied to the wideband jamming case by replacing ρ_N in (2-28) by ρ_T where

$$\rho_T \triangleq \frac{S}{\sigma_T^2} = \frac{S}{\sigma_N^2 + \sigma_J^2}. \tag{2-37}$$

Therefore, we have for the case of wideband jamming

$$P_b(e) = \frac{M}{2(M-1)} \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m+1}}{(m+1)^L} \sum_{j=0}^{m(L-1)} \frac{c_{j,m}}{(m+1)^j} \mathcal{L}_j^{(L-1)}\left(-\frac{L\rho_T}{m+1}\right) \tag{2-38}$$

where

$$c_{j,m} = \frac{1}{j} \sum_{q=1}^{j} \binom{j}{q} [(m+1)q-j]c_{j-q,m}b_q, \quad 1 \leq j \leq m(L-1) \tag{2-39a}$$

$$c_{0,m} = 1 \tag{2-39b}$$

with

$$b_q = \begin{cases} 1, & q \leq L-1 \\ 0, & q > L-1 \end{cases} . \tag{2-40}$$

2.3 PERFORMANCE OF SQUARE-LAW LINEAR COMBINING RECEIVER IN THE PARTIAL-BAND NOISE JAMMING CHANNEL

As is shown in Figure 2-9, we assume a fraction γ of the total bandwidth is jammed by a noise-like signal. We further assume that the jamming bandwidth is constrained to cover exactly a complete number of M-ary cluster locations, i.e. no matter where a cluster is hopped all M possible frequency selections will be either all jammed or all unjammed. This assumption is made to simplify considerably the analysis by eliminating edge effects of partially-jammed clusters at the edges of the jammed band. If $N \gg 1$, then the probability of hopping into such a partially-jammed cluster is small and the approximation due to ignoring edge effects is very good.

We assume the total jamming power, J, is distributed uniformly across a fraction γ of the total frequency cells, each of which has bandwidth B. The jamming power in a jammed cell is then given by

$$\sigma_J^2 = \frac{J}{\gamma W} B \quad \text{watts.} \tag{2-41}$$

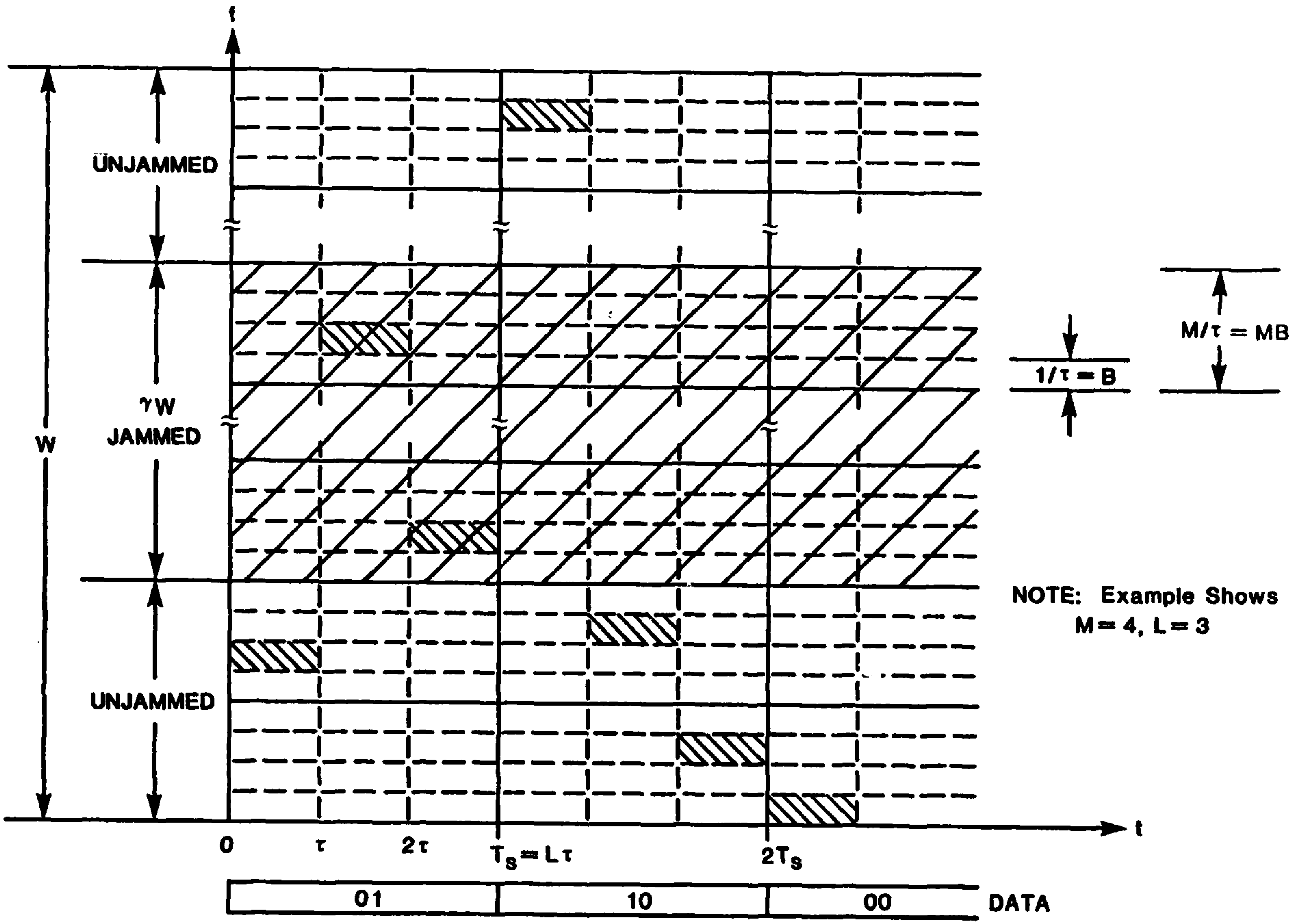


FIGURE 2-9 M-ARY FSK/FH SIGNAL IN PARTIAL-BAND JAMMING ENVIRONMENT

A specific M-ary cluster is received jamming-free with probability $1-\gamma$; and perturbed by jamming noise of power σ_J^2, with probability γ.

The dehopped received waveform r(t) may be represented during any given hop as:

$$r(t) = \begin{cases} s(t) + n(t) + j(t), & \text{with probability } \gamma \\ s(t) + n(t), & \text{with probability } 1-\gamma \end{cases} \tag{2-42}$$

where s(t) is the information-bearing signal, and n(t) and j(t) are thermal noise and jamming noise, respectively. Let us define an event J_e where $J_e=0$ denotes the absence of j(t) from r(t) and $J_e=1$ denotes the presence of j(t) in r(t) during any given hop:

$$J_e = \begin{cases} 1; \; j(t) \text{ is present in } r(t) \text{ with } \Pr(J_e = 1) = \gamma \\ 0; \; j(t) \text{ is absent from } r(t) \text{ with } \Pr(J_e = 0) = 1-\gamma. \end{cases} \tag{2-43}$$

We can further define the sequence of jamming events over the L hops which make up one M-ary symbol by the vector

$$\underline{J}_e = \left(J_{e1}, J_{e2}, \ldots, J_{eM}\right) \tag{2-44}$$

where components J_{ei}, $i = 1, 2, \ldots, M$, are jamming events as defined in (2-43).

Over a hop interval the signal s(t) at the output of the dehopping mixer in the receiver is

$$s(t) = \sqrt{2S} \cos\left(2\pi f_i t + \theta_i\right), \text{ symbol "i" transmitted,} \quad i = 1, 2, \ldots, M, \tag{2-45}$$

where S is the received (average) signal power; f_i is the frequency for the i-th symbol, $i = 1, 2, \ldots, M$; and θ_i, $i = 1, 2, \ldots, M$, are independent phases uniformly distributed on $[0, 2\pi)$.

Assuming that the thermal noise and the jamming noise in any selected cell are Gaussian distributed, we may express the $n_i(t)$ and $j_i(t)$, $i = 1, 2, \ldots, M$, at the outputs of the bandpass filters in the form of a Rician decomposition:

$$n_i(t) = n_{ci}(t) \cos 2\pi f_i t + n_{si}(t) \sin 2\pi f_i t; \; i = 1, 2, \ldots, M, \tag{2-46}$$

$$j_i(t) = j_{ci}(t) \cos 2\pi f_i t + j_{si}(t) \sin 2\pi f_i t; \; i = 1, 2, \ldots, M, \tag{2-47}$$

where $n_{ci}(t)$, $n_{si}(t)$, $j_{ci}(t)$, and $j_{si}(t)$ at a given time are statistically independent Gaussian random variables with variances (or average power) given by

$$E\left[n_i^2(t)\right] = E\left[n_{ci}^2(t)\right] = E\left[n_{si}^2(t)\right] = \sigma_N^2 \tag{2-48}$$

and

$$E\left[j_i^2(t)\right] = E\left[j_{ci}^2(t)\right] = E\left[j_{si}^2(t)\right] = \sigma_J^2. \tag{2-49}$$

Since $n_i(t)$ and $j_i(t)$ are additive noises the resultant noise power σ^2 at the inputs to the envelope detectors may be written as

$$\sigma^2 = \begin{cases} \sigma_N^2, & J_e = 0 \text{ with } \Pr(J_e = 0) = 1-\gamma \quad (2\text{-}50) \\ \sigma_T^2 = \sigma_N^2 + \sigma_J^2, & J_e = 1 \text{ with } \Pr(J_e = 1) = \gamma \quad (2\text{-}51) \end{cases}$$

where σ_N^2 is the thermal noise power as given by (2-48) and σ_J^2 is the jamming power given by (2-41).

The receiver shown in Figure 2-1 forms the squared envelopes of the outputs of the M bandpass filters and samples these envelopes once per hop for each of the L hops forming an M-ary symbol, thus forming the samples $z_{k\ell}$, $k = 1, 2, \ldots, M$, $\ell = 1, 2, \ldots, L$. The L samples from each channel are summed, forming the variables z_k, $k = 1, 2, \ldots, M$. By selecting the largest z_k the transmitted signal is identified. A correct decision that the i-th symbol was transmitted is made if $z_i > z_j$, $j \neq i$, for all $j \in \{1, 2, \ldots, M\}$. Otherwise an erroneous decision is made.

Since successive hops may be jammed or unjammed, the symbol error probability must be averaged over the possible jamming sequences $\underline{J}_e$, i.e.

$$P_s(e) = E_{\underline{J}_e}\left\{P_s(e|\underline{J}_e)\right\}. \tag{2-52}$$

Consider a jamming sequence $\underline{J}_e = (J_{e1}, J_{e2},\ldots,J_{eL})$ where

$$J_{ei} = \begin{cases} 1, & i = 1, 2,\ldots,\ell \\ 0, & i = \ell+1, \ell+2,\ldots,L, \end{cases} \tag{2-53}$$

i.e. the first ℓ hops are jammed and the remaining $L-\ell$ hops are unjammed. This gives rise to a certain set of decision variables $\{z_k\}$. Now consider another jamming sequence $\underline{J}_e'$ obtained by permuting the order of the J_{ei} given by (2-53) with the requirement that $\underline{J}_e' \neq \underline{J}_e$. This gives rise to a set of decision variables $\{z_k'\}$. Since the noise and the jamming processes are assumed to be stationary, the statistical properties of a jammed z_{ki} do not depend upon which hop out of the L hops it represents; and similarly for unjammed hops. Therefore the statistics of $\{z_k'\}$ are the same as those of $\{z_k\}$ and

$$P_s(e|\underline{J}_e') = P_s(e|\underline{J}_e), \tag{2-54}$$

i.e. the error probability depends only on the number of hops jammed and not on the order of their occurence in the pattern of jammed and unjammed hops forming an M-ary symbol. We may thus write (2-52) as

$$P_s(e) = \sum_{\ell=0}^{L} \binom{L}{\ell} \gamma^{\ell}(1-\gamma)^{L-\ell}\, P_s(e|\ell) \tag{2-55}$$

where $P_s(e|\ell)$ is a short-hand notation for P_s(e|exactly ℓ hops are jammed out of L hops sent). Furthermore, since a decision either is or is not correct,

$$P_s(e|\ell) = 1 - P_s(\text{correct}|\ell). \tag{2-56}$$

The probability given by (2-56) is the probability of making an incorrect symbol decision. The communicator, however, is more commonly interested in the probability of making a bit error. The conversion of symbol error probability to bit error probability is given by Lindsey and Simon [5, p. 198]:

$$P_b(e) = \frac{2^{K-1}}{2^K - 1} P_s(e) = \frac{M}{2(M-1)} P_s(e) \tag{2-57}$$

where $P_s(e)$ is given by (2-56).

We now proceed to develop an analytical expression for $P_s(\text{correct}|\ell)$. Without loss of generality we may assume that symbol "1" is transmitted. Then

$$P_s(\text{correct}|\ell) = \Pr\left\{z_1 > z_2 \text{ and } z_1 > z_3 \text{ and } \dots \text{ and } z_1 > z_M|\ell\right\}$$

$$= \int_0^\infty p_{z_1}(\zeta_1|\ell)\, P(z_2, z_3, \dots, z_M|z_1 = \zeta_1;\ell)d\zeta, \tag{2-58}$$

where $p_{z_1}(\zeta_1|\ell)$ is the conditional probability density function of z_1 given ℓ hops jammed and $P(z_2, z_3, \dots, z_M|z_1 = \zeta_1;\ell)$ is the conditional probability that all of $z_2, z_3, \dots, z_M$ are less than z_1 given that $z_1 = \zeta_1$ and ℓ hops are jammed.

Since the M cells of a cluster on any specific hop are either all jammed or all unjammed, the summed envelope samples z_i, $i = 2, 3, \dots, M$, are all identically distributed. Since, furthermore, the channels are independently distributed,

$$P(z_2, z_3, \dots, z_M|z_1 = \zeta_1;\ell) = P(z_2|z_1 = \zeta_1;\ell)\, P(z_3|z_1 = \zeta_1;\ell) \cdots P(z_M|z_1 = \zeta_1;\ell)$$

$$= \left[P(z|z_1 = \zeta_1;\ell)\right]^{M-1} \tag{2-59}$$

where $P(z|z_1 = \zeta_1;\ell)$ represents any one of the identical conditional distributions $P(z_i|z_1 = \zeta_1;\ell)$, $i = 2, 3,\ldots,M$. We may write

$$P(z_i|z_1 = \zeta_1;\ell) = \int_0^{\zeta_1} p_z(\zeta|\ell)d\zeta \tag{2-60}$$

where $p_z(\zeta|\ell)$ is the conditional probability density function of any one of the identically distributed z_i, $i = 2, 3,\ldots,M$, given that ℓ hops are jammed. From (2-58)-(2-60),

$$P_s(e|\ell) = 1 - \int_0^{\infty} p_{z_1}(\zeta_1|\ell)\left[\int_0^{\zeta_1} p_z(\zeta|\ell)d\zeta\right]^{M-1} d\zeta_1. \tag{2-61}$$

We now need to find the conditional probability density functions $p_z(\zeta|\ell)$ and $p_{z_1}(\zeta_1|\ell)$ and evaluate (2-61). In an effort to obtain a computationally useful form, we pursue two methods of evaluating (2-61): the characteristic function method and the direct method.

2.3.1 Characteristic Function Method

In our previous work [1], the characteristic function method proved useful in evaluating the bit error probability for FH/BFSK in the presence of partial-band noise jamming. Consequently we are motivated to approach the problem of FH/MFSK by the same method.

2.3.1.1 Probability Density Function of a No-Signal Channel

Consider first the channels in which signal is not present. It is well-known [6, Sec. 4.3] that the squared envelope of Gaussian noise is a scaled centrally chi-square distributed random variable with two degrees of freedom. Therefore, each z_{ki} is centrally chi-square distributed. However, the scaling of the jammed hops differs from the scaling of the unjammed hops, in accordance with the two possible total noise variances defined in (2-50) and (2-51).

Let Z_1 denote the sum of ℓ jammed-hop z_{ki}'s and Z_2 denote the sum of $L-\ell$ unjammed-hop z_{ki}'s. Since all the components which go to making up either Z_i, $i = 1, 2$, have identical variances, and therefore the same scaling, each Z_i, $i = 1, 2$, is scaled chi-square distributed with the degrees of freedom equal to the sum of the degrees of freedom of the components of the Z_i. Therefore, we can write the conditional probability density functions of the sum of the squared envelopes for Z_1 as

$$p_{Z_1}(\zeta_1) = \left(\frac{1}{2\sigma_T^2}\right)^{\ell} \frac{1}{\Gamma(\ell)} \zeta_1^{\ell-1} \exp\left(-\zeta_1/2\sigma_T^2\right), \quad \zeta_1 > 0 \tag{2-62a}$$

and the sum of the squared envelopes for Z_2 as

$$p_{Z_2}(\zeta_2) = \left(\frac{1}{2\sigma_N^2}\right)^{L-\ell} \frac{1}{\Gamma(L-\ell)} \zeta_2^{L-\ell} \exp\left(-\zeta_2/2\sigma_N^2\right), \quad \zeta_2 > 0. \tag{2-62b}$$

To find the probability density function for the sum $Z_1 + Z_2$, i.e. jammed hops plus unjammed hops, we turn to the characteristic function method and make use of the fact that the characteristic function for the sum of two random variables is the product of the characteristic functions of the two random variables.

The characteristic functions of the probability density functions given by (2-62) are, respectively,

$$\Phi_1(j\nu) = \frac{1}{(1 - j2\sigma_T^2)^{\ell}} \tag{2-63a}$$

and

$$\Phi_2(j\nu) = \frac{1}{(1 - j2\sigma_N^2)^{L-\ell}} . \tag{2-63b}$$

Therefore we require the distribution of $Z_1 + Z_2$ corresponding to the characteristic function

$$\Phi(j\nu) \triangleq \Phi_1(j\nu)\Phi_2(j\nu) = \frac{1}{(1-j2\sigma_T^2\nu)^{\ell}\ (1-j2\sigma_N^2\nu)^{L-\ell}} . \tag{2-64}$$

The density function corresponding to $\Phi(j\nu)$ may be obtained by taking the inverse Fourier transform of (2-64); but to do this we need a partial fraction expansion for the right-hand side of (2-64). As shown in Appendix 2A, the required partial fraction expansion is

$$\Phi(j\nu) = (-1)^{\ell}\left(\frac{1}{2\sigma_T^2}\right)^{\ell}\left(\frac{1}{2\sigma_N^2}\right)^{L-\ell}\left[\sum_{r=0}^{\ell}\frac{(-1)^r(2\sigma_T^2)^r A_r}{(1-j2\sigma_T^2\nu)^r} + \sum_{r=0}^{L-\ell}\frac{(-1)^r(2\sigma_N^2)^r B_r}{(1-j2\sigma_N^2\nu)^r}\right] \tag{2-65}$$

where

$$A_0 = B_0 = 0, \tag{2-66a}$$

$$A_r = \frac{(-1)^{\ell-r}(L-\ell)_{\ell-r}}{(\ell-r)!}\left(\frac{1}{1-\delta}\right)^{L-r}\left(2\sigma_T^2\right)^{L-r}, \quad r = 1, 2, \ldots, \ell, \tag{2-66b}$$

and

$$B_r = \frac{(-1)^{L-\ell-r}(\ell)_{L-\ell-r}}{(L-\ell-r)!}\left(\frac{1}{\delta-1}\right)^{L-r}\left(2\sigma_T^2\right)^{L-r}, \quad r = 1, 2, \ldots L-\ell, \tag{2-66c}$$

with the parameter

$$\delta \triangleq \sigma_T^2/\sigma_N^2 \tag{2-67}$$

and where the Pochhammer symbol is defined [4, eq. 6.1.22] by

$$(a)_0 = 1 \tag{2-68a}$$

$$(a)_n = \Gamma(a+n)/\Gamma(n). \tag{2-68b}$$

We now make use of the relation [6, p. 110]

$$\mathcal{F}^{-1}\left\{(1-2j\omega)^{-n/2}\right\} = \frac{\zeta^{(n-2)/2}e^{-\zeta/2}}{2^{n/2}\Gamma(n/2)} \tag{2-69}$$

to take the inverse transform of (2-65) to obtain the desired conditional probability density function

$$p_z(\zeta|\ell) = (-1)^L \left(\frac{1}{2\sigma_T^2}\right)^{\ell}\left(\frac{1}{2\sigma_N^2}\right)^{L-\ell}\left[\sum_{r=0}^{\ell} (-1)^r A_r \frac{1}{\Gamma(r)} \zeta^{r-1} e^{-\zeta/2\sigma_T^2}\right.$$

$$\left. + \sum_{r=0}^{L-\ell} (-1)^r B_r \frac{1}{\Gamma(r)} \zeta^{r-1} e^{-\zeta/2\sigma_N^2}\right] \tag{2-70}$$

where the coefficients A_r and B_r are given by (2-66).

2.3.1.2 Probability Density Function of the Signal Channel

We now turn our attention to the probability density function of the signal channel. We define the signal-to-total noise ratio for a jammed hop as

$$\rho_T \triangleq S/\sigma_T^2, \tag{2-71a}$$

and the signal-to-thermal noise ratio for an unjammed hop as

$$\rho_N \triangleq S/\sigma_N^2. \tag{2-71b}$$

We also define the signal-to-jamming ratio for a jammed hop

$$\rho_J \triangleq S/\sigma_J^2. \tag{2-71c}$$

Then we may also write (2-71a) as

$$\rho_T = \left(\rho_N^{-1} + \rho_J^{-1}\right)^{-1}. \tag{2-71d}$$

The squared envelope of a sine wave in Gaussian noise is known to be a scaled noncentral chi-square variable with 2 degrees of freedom and noncentral

parameter λ equal to twice the power SNR [6, Sec. 4.7]. The sum of ℓ such envelopes with the same scaling will be a scaled noncentral chi-square variable with 2ℓ degrees of freedom and noncentral parameter $\lambda = 2\ell$(power SNR). Thus, the conditional density function for the sum of ℓ jammed envelope samples is

$$p_1(\xi_1|\ell) = \frac{1}{2\sigma_T^2}\left(\frac{\xi_1}{4\sigma_T^2\ell\rho_T}\right)^{(\ell-1)/2} I_{\ell-1}\left(\sqrt{\frac{4\ell\rho\xi_1}{\sigma_T^2}}\right)\exp\left(-\frac{\xi_1}{2\sigma_T^2} - 2\ell\rho_T\right), \quad \xi_1 > 0 \tag{2-72}$$

and the conditional density function for the sum of L-ℓ unjammed envelope samples is

$$p_2(\xi_2|\ell) = \frac{1}{2\sigma_N^2}\left(\frac{\xi_2}{4\sigma_N^2(L-\ell)\rho_N}\right)^{(L-\ell-1)/2} I_{L-\ell-1}\left(\sqrt{\frac{4(L-\ell)\xi_2\rho_N}{\sigma_N^2}}\right)$$

$$\cdot \exp\left[-\frac{\xi_2}{2\sigma_N^2} - 2(L-\ell)\rho_N\right], \qquad \xi_2 > 0. \tag{2-73}$$

Then the required density for the sum of the jammed and the unjammed hops may be found by taking the inverse Fourier transform of the product of the characteristic functions of p_1 and p_2.

The characteristic functions of the densities given by (2-72) and (2-73) are, respectively,

$$\Psi_1(j\nu) = \frac{e^{-\ell\rho_T}}{(1 - j2\sigma_T^2\nu)^{\ell}} \exp\left(\frac{\ell\rho_T}{1 - j2\sigma_T^2\nu}\right) \tag{2-74}$$

and

$$\Psi_2(j\nu) = \frac{e^{-(L-\ell)\rho_N}}{(1 - j2\sigma_N^2\nu)^{L-\ell}} \exp\left[\frac{(L-\ell)\rho_N}{1 - j2\sigma_N^2\nu}\right]. \tag{2-75}$$

If we let $\Psi(j\nu)$ denote the characteristic function of $p_{z_1}(\zeta_1)$, we have

$$\Psi(j\nu) = \Psi_1(j\nu)\Psi_2(j\nu). \tag{2-76}$$

In order to facilitate the taking of an inverse Fourier transform of $\Psi(j\nu)$, we use (2-74) and (2-75) in (2-76) and expand the exponentials involving $j\nu$ into Taylor series. After some regrouping of terms, we obtain

$$\Psi(j\nu) = e^{-\ell\rho_T} e^{-(L-\ell)\rho_N} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(\ell\rho_T)^m}{m!} \frac{\left[(L-\ell)\rho_N\right]^n}{n!} \Psi_{m,n}(j\nu) \tag{2-77}$$

where

$$\Psi_{m,n}(j\nu) = \frac{1}{(1 - j2\sigma_T^2\nu)^{\ell+m}(1 - j2\sigma_N^2\nu)^{L-\ell+n}} \cdot \tag{2-78}$$

We now require a partial fraction of $\psi_{m,n}(j\nu)$ as given in (2-78) in order to obtain forms whose inverse Fourier transforms are known. The required partial fraction expansion is developed in Appendix 2B. Upon substituting the results from (2B-14) into (2-77) we obtain

$$\Psi(j\nu) = e^{-\ell\rho_T} e^{-(L-\ell)\rho_N} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(\ell\rho_T)^m \left[(L-\ell)\rho_N\right]^n}{m!n!} (-1)^{L+m+n} \left(\frac{1}{2\sigma_T^2}\right)^{\ell+m} \left(\frac{1}{2\sigma_N^2}\right)^{L-\ell+n}$$

$$\cdot \left[\sum_{r=0}^{\ell+m} C_r \frac{(-1)^r(2\sigma_T^2)^r}{(1 - j2\sigma_T^2\nu)^r} + \sum_{r=0}^{L-\ell+n} D_r \frac{(-1)^r(2\sigma_N^2)^r}{(1 - j2\sigma_N^2\nu)^r}\right] \tag{2-79}$$

where

$$C_0 = D_0 = 0, \tag{2-80a}$$

$$C_r = \frac{(-1)^{L-\ell+n}(L-\ell+n)_{\ell+m-r}}{(\ell+m-r)!}\left(\frac{2\sigma_T^2}{\delta-1}\right)^{L+m+n-r}, \quad r=1,\,2,\ldots,\ell+m, \tag{2-80b}$$

and

$$D_r = \frac{(-1)^{L-\ell+n-r}(\ell+m)_{L-\ell+n-r}}{(L-\ell+n-r)!}\left(\frac{2\sigma_T^2}{\delta-1}\right)^{L+m+n-r}, \quad r=1,\,2,\ldots,L-\ell+n \tag{2-80c}$$

with δ as defined in (2-67).

The inverse Fourier transform of (2-79) is the required probability density function $p_{z_1}(\zeta_1|\ell)$. In order to simplify the result, we introduce the term

$$(1-\delta_{r,0}) = \begin{cases} 0, & r=0 \\ 1, & r\neq 0, \end{cases} \tag{2-81}$$

where $\delta_{r,0}$ is the Kronecker delta function*, into (2-80b) and (2-80c) in order to account for the cases of (2-80a) without having to split out certain cases of the summations in (2-79) for which $\ell+m$ or $L-\ell+n$ may be zero. With this modification, we can use (2-69) to take the inverse Fourier transform of (2-79). After substituting (2-80) into the result and a bit of algebraic simplification and regrouping, we have the result

*The ratio δ defined in (2-67) is distinguished from the Kronecker delta function by the absence of subscripts on the former.

$$p_{z_1}(\zeta_1|\ell) = \frac{1}{2\sigma_T^2} e^{-\ell\rho_T} e^{-(L-\ell)\rho_N} \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} \frac{(\ell\rho_T)^m \left[(L-\ell)\rho_N\right]^n}{m!n!}$$

$$\cdot \left[e^{-\zeta_1/2\sigma_T^2} \sum_{r=0}^{\ell+m} \left(1-\delta_{r,0}\right) \frac{(-1)^{\ell+m-r}(L-\ell+n)_{\ell+m-r}}{(\ell+m-r)!} \frac{\delta^{L-\ell+n}}{\Gamma(r)} \right.$$

$$\cdot \left(\frac{1}{\delta-1}\right)^{L+m+n-r} \left(\frac{\zeta_1}{2\sigma_T^2}\right)^{r-1}$$

$$+ e^{-\delta\zeta_1/2\sigma_T^2} \sum_{r=0}^{L-\ell+n} \left(1-\delta_{r,0}\right) \frac{(-1)^{\ell+m}(\ell+m)_{L-\ell+n-r}}{(L-\ell+n-r)!} \frac{\delta^{L-\ell+n}}{\Gamma(r)}$$

$$\left. \cdot \left(\frac{1}{\delta-1}\right)^{L+m+n-r} \left(\frac{\delta\zeta_1}{2\sigma_T^2}\right)^{r-1} \right] . \qquad (2\text{-}82)$$

2.3.1.3 Probability of Error

Now that we have (2-70) for $p_z(\zeta|\ell)$ and (2-82) for $p_{z_1}(\zeta_1|\ell)$, we seek to evaluate $P_s(\text{correct}|\ell)$ as defined by the integral in (2-61). The first step is to evaluate $P(z|z_1 = \zeta_1;\ell)$ as indicated in (2-60). We substitute (2-70) into (2-60), interchange the order of integration and summation, and make the changes of variables $u = \zeta/2\sigma_T^2$ in the first integral and $v = \zeta/2\sigma_N^2$ in the second second integral. This yields the form

$$P(z|z_1=\zeta_1;\ell) = (-1)^L\left(\frac{1}{2\sigma_T^2}\right)^{\ell}\left(\frac{1}{2\sigma_N^2}\right)^{L-\ell}\left[\sum_{r=0}^{\infty}\left(1-\delta_{r,0}\right)(-1)^r\,\frac{A_r(2\sigma_T^2)^r}{\Gamma(r)}\int_0^{\zeta_1/2\sigma_T^2} u^{r-1}\,e^{-u}\,du\right.$$

$$\left.+\sum_{r=0}^{L-\ell}\left(1-\delta_{r,0}\right)(-1)^r\,\frac{B_r(2\sigma_N^2)^r}{\Gamma(r)}\int_0^{\zeta_1/2\sigma_N^2} v^{r-1}\,e^{-v}\,dv\right]. \tag{2-83}$$

The integrals in (2-83), with the inclusion of the factor $1/\Gamma(r)$, are now recognized as one form of the incomplete gamma function [4, eq. 6.5.1]

$$P(a,x) = \frac{1}{\Gamma(a)}\int_0^x t^{a-1}\,e^{-t}\,dt. \tag{2-84}$$

Using the definition of δ in (2-67) and the coefficients A_r and B_r from (2-66), we can substitute (2-84) into (2-83) and, after some algebraic simplifications, arrive at the form

$$P(z|z_1=\zeta_1;\ell) = \sum_{r=0}^{\ell}\left(1-\delta_{r,0}\right)\frac{(-1)^{\ell-r}(L-\ell)_{\ell-r}}{(\ell-r)!}\left(\frac{1}{\delta-1}\right)^{L-r}\delta^{L-\ell}P\left(r,\frac{\zeta_1}{2\sigma_T^2}\right)$$

$$+\sum_{r=0}^{L-\ell}\left(1-\delta_{r,0}\right)\frac{(-1)^{\ell}(\ell)_{L-\ell-r}}{(L-\ell-r)!}\left(\frac{\delta}{\delta-1}\right)^{L-r}\left(\frac{1}{\delta}\right)^{\ell}P\left(r,\frac{\delta\zeta_1}{2\sigma_T^2}\right). \tag{2-85}$$

We now combine (2-60), (2-61), (2-82), and (2-85) to obtain an expression for the probability of making a symbol error. We make use of the fact that

$$\sum_{\ell=0}^{L}\binom{L}{\ell}\gamma^{\ell}(1-\gamma)^{L-\ell} = 1, \tag{2-86}$$

interchange the order of integration and summation over m and n, make the change of variable $x = \xi_1/2\sigma_T^2$, factor $(-1)^\ell$ from each of the summations, and use $(-1)^{-r} = (-1)^r$ to obtain the result

$$P_s(e) = 1 - \sum_{\ell=0}^{L} (-1)^{\ell M} \binom{L}{\ell} \gamma^\ell (1-\gamma)^{L-\ell} e^{-\ell\rho_T} e^{-(L-\ell)\rho_N} \sum_{m=0}^{\infty} \frac{(\ell\rho_T)^m}{m!} \sum_{n=0}^{\infty} \frac{[(L-\ell)\rho_N]^n}{n!}$$

$$\cdot \int_0^\infty \left\{ \sum_{r=0}^{\ell+m} \left(1-\delta_{r,0}\right) \frac{(-1)^{m-r}(L-\ell+n)_{\ell+m-r}}{(\ell+m-r)!} \delta^{L-\ell+n} \left(\frac{1}{\delta-1}\right)^{L+m+n-r} \frac{1}{\Gamma(r)} x^{r-1} e^{-x} \right.$$

$$\left. + \sum_{r=0}^{L-\ell+n} \left(1-\delta_{r,0}\right) \frac{(-1)^m(\ell+m)_{L-\ell+n-r}}{(L-\ell+n-r)!} \delta^{L-\ell+n} \left(\frac{1}{\delta-1}\right)^{L+m+n-r} \frac{1}{\Gamma(r)} x^{r-1} e^{-\delta x} \right\}$$

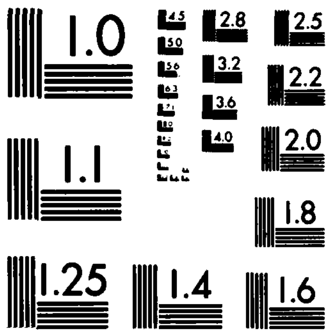
$$\cdot \left\{ \sum_{r=0}^{\ell} \left(1-\delta_{r,0}\right) \frac{(-1)^r(L-\ell)_{\ell-r}}{(\ell-r)!} \delta^{L-\ell} \left(\frac{1}{\delta-1}\right)^{L-r} P(r,x) \right.$$

$$\left. + \sum_{r=0}^{L-\ell} \left(1-\delta_{r,0}\right) \frac{(\ell)_{L-\ell-r}}{(L-\ell-r)!} \left(\frac{1}{\delta-1}\right)^{L-r} \delta^{L-\ell-r} P(r,\delta x) \right\}^{M-1} dx. \qquad (2\text{-}87)$$

Equation (2-87) still contains one integral to be evaluated. However, considering the complicated form of the integrand which includes powers of sums of incomplete gamma functions, one can easily expect that the explicit representation of the result of performing this integration will probably be so cumbersome as to be, at best, of academic interest only. This, indeed, is the case as is evident from the result in Appendix 2C wherein we

AD-A147 766 OPTIMUM JAMMING EFFECTS ON FREQUENCY-HOPPING M-ARY FSK 2/9
(FREQUENCY-SHIFT K..(U) LEE (J S) ASSOCIATES INC
ARLINGTON VA J S LEE ET AL. OCT 84 JC-2025-N

UNCLASSIFIED N00014-83-C-0312 F/G 17/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

pursue the further formal evaluation of the integral in (2-87). From a practical viewpoint, even (2-87) is exceedingly difficult to evaluate numerically, and therefore we seek an alternative formulation. However, (2-87) can be reduced to useful forms in certain special cases, which are discussed in Appendix 2D.

2.3.2 Direct Method

As an alternative to the characteristic function method, we may employ a direct method to obtain the probability density functions of z_1 and z in (2-61). We use the result from Appendix 2E to write

$$p_{z_1}(\zeta_1|\ell) = \frac{1}{2\delta^\ell} \exp\left(-\frac{\lambda_\ell + \zeta_1}{2}\right) \sum_{k=0}^{\infty} \sum_{r=0}^{\infty} \frac{1}{k!r!} \left(\frac{\lambda_{0,\ell}}{2}\right)^k \left(\frac{\lambda_{1,\ell}}{2\delta}\right)^r \frac{(\zeta_1/2)^{k+r+L-1}}{\Gamma(k+r+L)}$$

$$\cdot \, {}_1F_1\left(r+\ell;\ k+r+L;\ \frac{(\delta-1)\zeta_1}{2\delta}\right) \tag{2-88}$$

and

$$p_z(\zeta) = \frac{1}{2\delta^\ell} \exp\left(-\frac{\zeta}{2\delta}\right) \left(\frac{\zeta}{2}\right)^{L-1} \frac{1}{\Gamma(L)}\, {}_1F_1\left(\ell;\ L;\ \frac{(\delta-1)\zeta}{2\delta}\right) \tag{2-89}$$

where δ is as defined earlier in (2-67),

$$\lambda_{0,\ell} = \frac{2(L-\ell)E_s}{LN_0} = \frac{2(L-\ell)S}{L\,N_0B} = 2(L-\ell)\rho_N, \tag{2-90a}$$

$$\lambda_{1,\ell} = \frac{2\ell E_s}{LN_T} = \frac{2\ell S}{LN_TB} = 2\ell\rho_T, \tag{2-90b}$$

and

$$\lambda_\ell = \lambda_{0,\ell} + \lambda_{1,\ell} \,. \tag{2-90c}$$

We use (2-89) to evaluate the inner integral in (2-61):

$$\int_0^{\zeta_1} p_z(\zeta)d\zeta = 1 - \int_{\zeta_1}^{\infty} p_z(\zeta)d\zeta$$

$$= 1 - \frac{1}{\delta^L\,\Gamma(L)} \int_{\zeta_1/2}^{\infty} e^{-x}\, x^{L-1}\, {}_1F_1\!\left(\ell;\, L;\, \frac{\delta-1}{\delta}\, x\right) \qquad (2\text{-}91)$$

where we have made the change of variable $x=\zeta/2$. To evaluate (2-91), we replace the confluent hypergeometric function by its series representation and interchange the order of summation and integration. The result is

$$P(z_i|z_1 = \zeta_1;\ell) = 1 - \frac{1}{\delta^\ell} \sum_{p=0}^{\infty} \left(\frac{\delta-1}{\delta}\right)^p \frac{(\ell)_p}{p!} \frac{\Gamma(L+p,\, \zeta_1/2)}{\Gamma(L+p)} \qquad (2\text{-}92)$$

where $\Gamma(\cdot,\cdot)$ is an incomplete gamma function [4, eq. 6.5.3].

We now use (2-92) to evaluate

$$[P(z_i|z_1 = \zeta_1;\ell)]^{M-1} = \sum_{m=0}^{M-1} \binom{M-1}{m} \left[- \frac{1}{\delta^\ell} \sum_{p=0}^{\infty} \left(\frac{\delta-1}{\delta}\right)^p \frac{(\ell)_p}{p!} \frac{\Gamma(L+p,\zeta_1/2)}{\Gamma(L+p)} \right]^m \qquad (2\text{-}93)$$

where we have applied the binomial theorem to the power of the right-hand side of (2-92). Using the relation [4, eq. 6.5.2, 6.5.3, and 6.5.13]

$$\frac{\Gamma(L+p,\zeta_1/2)}{\Gamma(L+p)} = e^{-\zeta_1/2} \sum_{q=0}^{L+p+1} \frac{(\zeta_1/2)^q}{q!}\,, \qquad (2\text{-}94)$$

we want to manipulate (2-93) into the form

$$[P(z_i|z_1 = \zeta_1;\ell)]^{M-1} = \sum_{m=0}^{M-1} (-1)^m \sum_{n=0}^{\infty} \frac{c_{nm}}{m!} \left(\frac{\zeta_1}{2}\right)^n . \qquad (2\text{-}95)$$

To accomplish this, we use the following development:

$$(1-A)^{\ell} \sum_{p=0}^{\infty} A^p \frac{(\ell)_p}{p!} \sum_{q=0}^{L+p-1} \frac{B^q}{q!} = \sum_{q=0}^{L-2} \frac{B^q}{q!} + \sum_{q=L-1}^{\infty} \frac{B^q}{q!} (1-A)^{\ell} \sum_{p=q-L+1}^{\infty} A^p \frac{(\ell)_p}{p!}$$

$$= \sum_{q=0}^{\infty} \frac{\beta_q}{q!} \left(\frac{\zeta_1}{2}\right)^q \tag{2-96}$$

where

$$A \triangleq \frac{\delta-1}{\delta} , \tag{2-97a}$$

$$B \triangleq \frac{\zeta_1}{2} , \tag{2-97b}$$

and

$$\beta_q = \begin{cases} 1, & 0 \leq q \leq L-1 \\ 1 - \dfrac{1}{\delta^{\ell}} \displaystyle\sum_{p=0}^{q-L} \left(\frac{\delta-1}{\delta}\right)^p \frac{(\ell)_p}{p!} , & q > L-1 . \end{cases} \tag{2-97c}$$

Now we must find the coefficients c_{nm} in (2-95) from the relation

$$\sum_{n=0}^{\infty} \frac{c_{nm}}{n!} \left(\frac{\zeta_1}{2}\right)^n = \left[\sum_{q=0}^{\infty} \frac{\beta_q}{q!} \left(\frac{\zeta_1}{2}\right)^q\right]^m . \tag{2-98}$$

We use the J.C.P. Miller formula [3, p. 42], just as we did in (2-19)-(2-22), to obtain the recurrence relation defining the coefficients:

$$c_{0m} = 1 \tag{2-99a}$$

$$c_{nm} = \frac{1}{n} \sum_{q=1}^{n} \binom{n}{q} [(m+1)q-n] c_{n-q,m} \beta_q . \tag{2-99b}$$

Substitution of (2-94), (2-96), (2-97), and (2-98) into (2-93) yields

$$[P(z|z_1 = \zeta_1;\ell)]^{M-1} = 1 - \sum_{m=1}^{M-1} \binom{M-1}{m} (-1)^{m+1} e^{-m\zeta_1/2} \sum_{n=0}^{\infty} \frac{c_{nm}}{n!} \left(\frac{\zeta_1}{2}\right)^n . \tag{2-100}$$

Further substitution of (2-100) into (2-61) yields the result

$$P_s(e|\ell) = \sum_{m=1}^{M-1} \binom{M-1}{m} (-1)^{m+1} \sum_{n=0}^{\infty} \frac{c_{nm}}{n!} \int_0^{\infty} e^{-m\zeta_1/2} \left(\frac{\zeta_1}{2}\right)^n p_{z_1}(\zeta_1) d\zeta_1 . \tag{2-101}$$

Using the probability density function $p_{z_1}(\zeta_1)$ given by (2-88), the integral in (2-101) becomes

$$\int_0^{\infty} e^{-m\zeta_1/2} \left(\frac{\zeta_1}{2}\right)^n p_{z_1}(\zeta_1)\, d\zeta_1 = \frac{1}{\delta^{\ell}} e^{-\lambda_\ell/2} \sum_{k=0}^{\infty} \sum_{r=0}^{\infty} \frac{1}{k!r!} \left(\frac{\lambda_{0,\ell}}{2}\right)^k \left(\frac{\lambda_{1,\ell}}{2}\right)^r \frac{1}{\Gamma(k+r+L)}$$

$$\cdot \int_0^{\infty} e^{-(m+1)x}\, x^{k+r+n+L-1}\, {}_1F_1\left(r+\ell;\, k+r+L;\, \frac{\delta-1}{\delta} x\right). \tag{2-102}$$

The integral on the right-hand side of (2-102) may be solved using [2, 7.621.4] to give

$$\int_0^{\infty} e^{-(m+1)x}\, x^{k+r+n+L-1}\, {}_1F_1\left(r+\ell;\, k+r+L;\, \frac{\delta-1}{\delta}\, x\right) = \frac{\Gamma(k+r+n+L)}{(m+1)^{k+r+n+L}}$$

$$\cdot\ {}_2F_1\left(r+\ell,\, k+r+n+L;\, k+r+L;\, \frac{\delta-1}{\delta(m+1)}\right) . \tag{2-103}$$

The hypergeometric function on the right-hand side of (2-103) can be manipulated to yield a finite sum [4, eq. 15.3.4]:

$$
{}_2F_1\left(r+\ell,\ k+r+n+L;\ k+r+L;\ \frac{\delta-1}{\delta(m+1)}\right) = \left[1-\frac{\delta-1}{\delta(m+1)}\right]^{-r-\ell} {}_2F_1\left(r+\ell,\ -n;\ k+r+L;\ \frac{\frac{\delta-1}{\delta(m+1)}}{\frac{\delta-1}{\delta(m+1)}-1}\right)
$$

$$
= \left[\frac{\delta(m+1)}{\delta m+1}\right]^{r+\ell} \sum_{j=0}^{n} \binom{n}{j}\left(\frac{\delta-1}{\delta m+1}\right)^{j} \frac{(r+\ell)_j}{(k+r+L)_j} \ . \tag{2-104}
$$

Finally, then, substitution of (2-102)-(2-104) into (2-101) yields the conditional error probability

$$
P_s(e|\ell) = \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m+1}}{(m+1)^{L-\ell}(\delta m+1)^{\ell}} \sum_{n=0}^{\infty} \frac{c_{nm}}{(m+1)^n n!}\, e^{-\lambda_\ell/2}
$$

$$
\cdot \sum_{k=0}^{\infty}\sum_{r=0}^{k} \frac{1}{(k-r)!\,r!} \left[\frac{\lambda_{0,\ell}}{2(m+1)}\right]^{k-r} \left[\frac{\lambda_{1,\ell}}{2(\delta m+1)}\right]^{r} (k+L)_n\, d_{nrk} \tag{2-105}
$$

where

$$
d_{nrk} = \sum_{j=0}^{n} \binom{n}{j}\left(\frac{\delta-1}{\delta m+1}\right)^{j} \frac{(r+\ell)_j}{(k+L)_j} \ . \tag{2-106}
$$

Putting the results from (2-105) and (2-106) into (2-55), and using (2-90) and (2-57), we obtain the unconditional error probability

$$
P_b(e) = \frac{M}{2(M-1)} \sum_{\ell=0}^{L} \binom{L}{\ell} \gamma^{\ell}(1-\gamma)^{L-\ell} \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m+1}}{(m+1)^{L-\ell}(\delta m+1)^{\ell}} \sum_{n=0}^{\infty} \frac{c_{nm}}{(m+1)^n n!}
$$

$$
\cdot \exp\left[-(L-\ell)\rho_N - \ell\rho_T\right] \sum_{k=0}^{\infty}\sum_{r=0}^{k} \frac{1}{(k-r)!\,r!} \left[\frac{(L-\ell)\rho_N}{m+1}\right]^{k-r}
$$

$$\cdot \left(\frac{\ell\rho_T}{\delta m+1}\right)^r (k+L)_n \sum_{j=0}^{n} \binom{n}{j} \left(\frac{\delta-1}{\delta m+1}\right)^j \frac{(r+\ell)_j}{(k+L)_j} \qquad (2\text{-}107)$$

where δ is defined in (2-67), ρ_N and ρ_T are defined in (2-71), and the coefficients c_{nm} are defined in (2-99).

2.4 NUMERICAL RESULTS FOR THE SQUARE-LAW LINEAR COMBINING RECEIVER

The numerical computations for wideband jamming (γ=1) are readily accomplished using (2-38) or (2D-4). However, the results for the general case of partial-band jamming are a much more difficult computational problem. Therefore, we discuss these two case separately.

2.4.1 Numerical Results for Wideband Jamming

We have computed (2D-4) by numerical integration for several cases of E_b/N_J, E_b/N_0, L, and M. In selecting values of E_b/N_0, we have chosen those values for which the error probability in the absence of jamming is equal to 10^{-3}, 10^{-4}, and 10^{-5} for ideal MFSK. Since the performance of ideal MFSK depends upon M, we must use the values of E_b/N_0 appropriate to the value of M being considered. The computations were performed using the values of E_b/N_0 to four decimal places as given in Table 2-1. However, the legends in the figures are rounded to two decimal places to reduce the size of the legends.

Figures 2-10 through 2-12 show the performance under wideband jamming of the square-law linear combining receiver for 4-ary FSK/FH with L=1, 4, and 6 hops/symbol, respectively, with the signal-to-thermal noise ratio E_b/N_0 as a parameter. As seen in Figure 2-10, the selected values of E_b/N_0 correspond to bit error rates of 10^{-3}, 10^{-4}, and 10^{-5} for the case of 1 hop/bit in the thermal-noise-limited region of the performance curve. From Figure 2-10 we see that the jamming becomes essentially negligible for $E_b/N_J \geq 30$ dB. From Figures 2-11 and 2-12 we see the degradation of

TABLE 2-1

VALUES OF ENERGY PER BIT TO NOISE DENSITY RATIO FOR WHICH BIT ERROR PROBABILITY OF MFSK IS EQUAL TO A GIVEN VALUE

M	E_b/N_0 for $P_b(e) = 1.000 \times 10^{-n}$ where n =		
	3	4	5
2	10.9444 dB	12.3133 dB	13.3525 dB
4	8.3524 dB	9.6284 dB	10.6065 dB
8	6.9718 dB	8.1690 dB	9.0939 dB
16	6.0696 dB	7.1996 dB	8.0783 dB
32	5.4183 dB	6.4910 dB	7.3295 dB

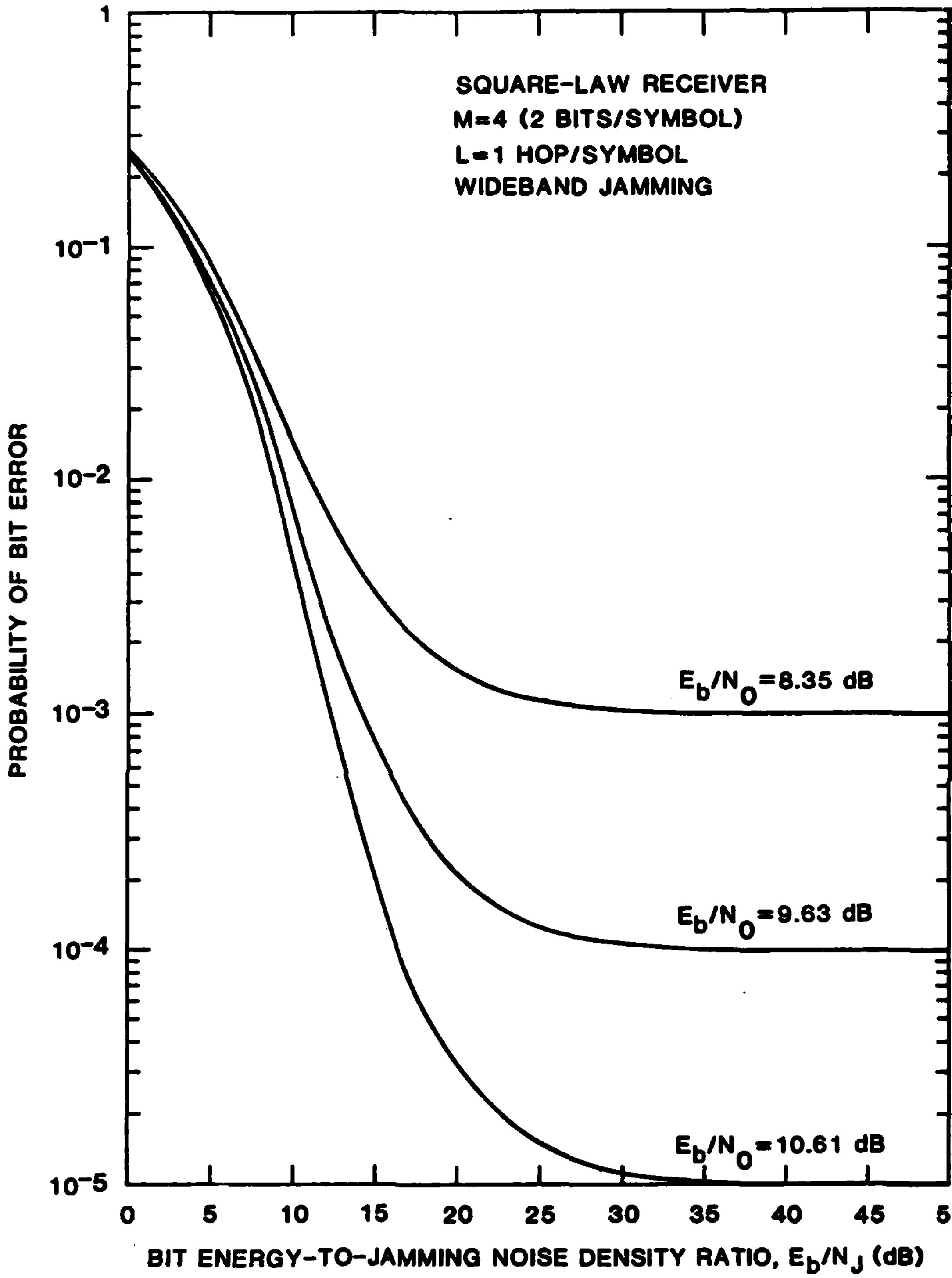


FIGURE 2-10 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK (M=4) SQUARE-LAW RECEIVER FOR L=1 HOP/SYMBOL WITH SIGNAL-TO-THERMAL NOISE RATIO (E_b/N_0) AS A PARAMETER

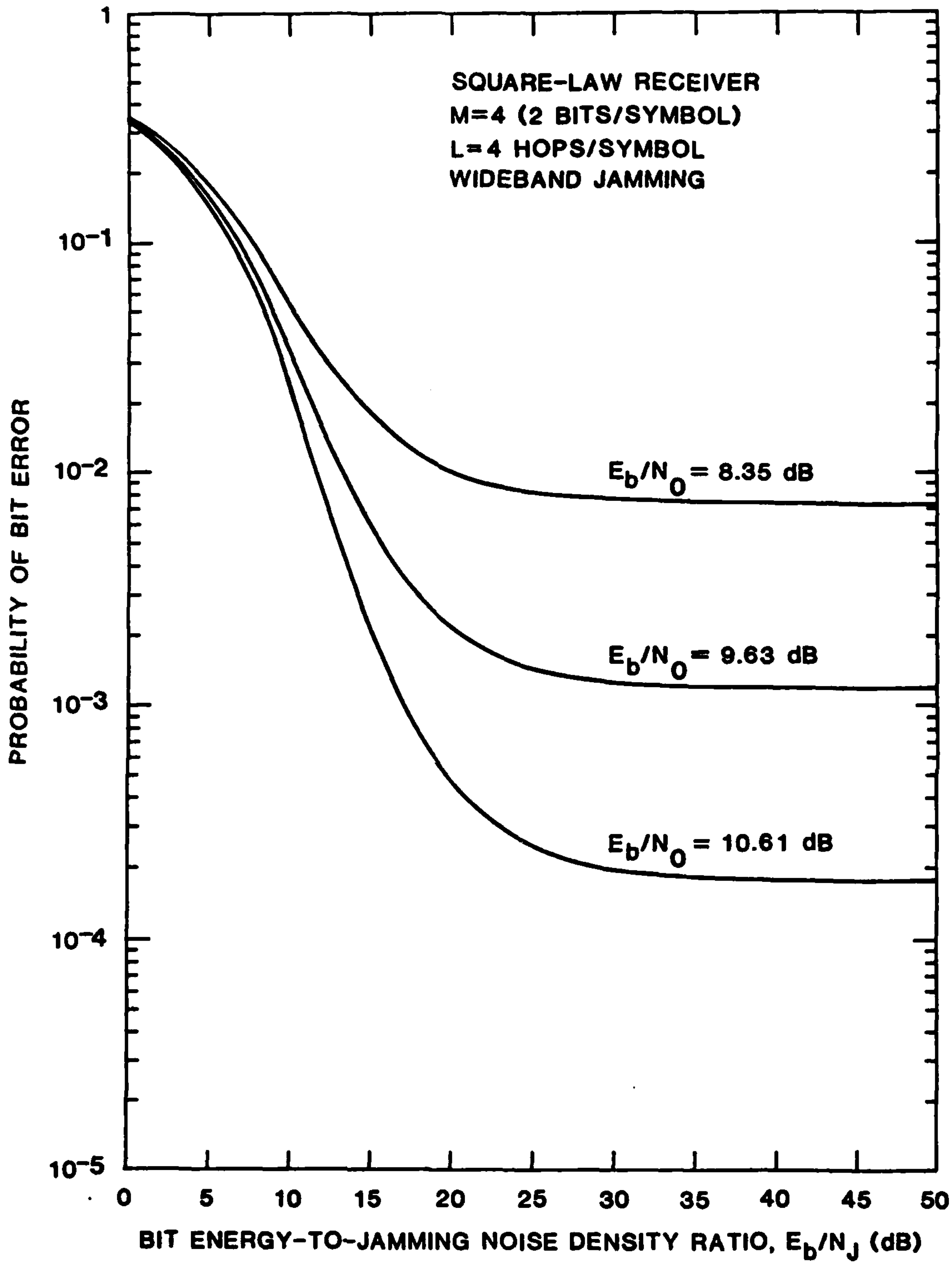


FIGURE 2-11 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK (M=4) SQUARE-LAW RECEIVER FOR L=4 HOPS/SYMBOL WITH SIGNAL-TO-THERMAL NOISE RATIO (E_b/N_0) AS A PARAMETER

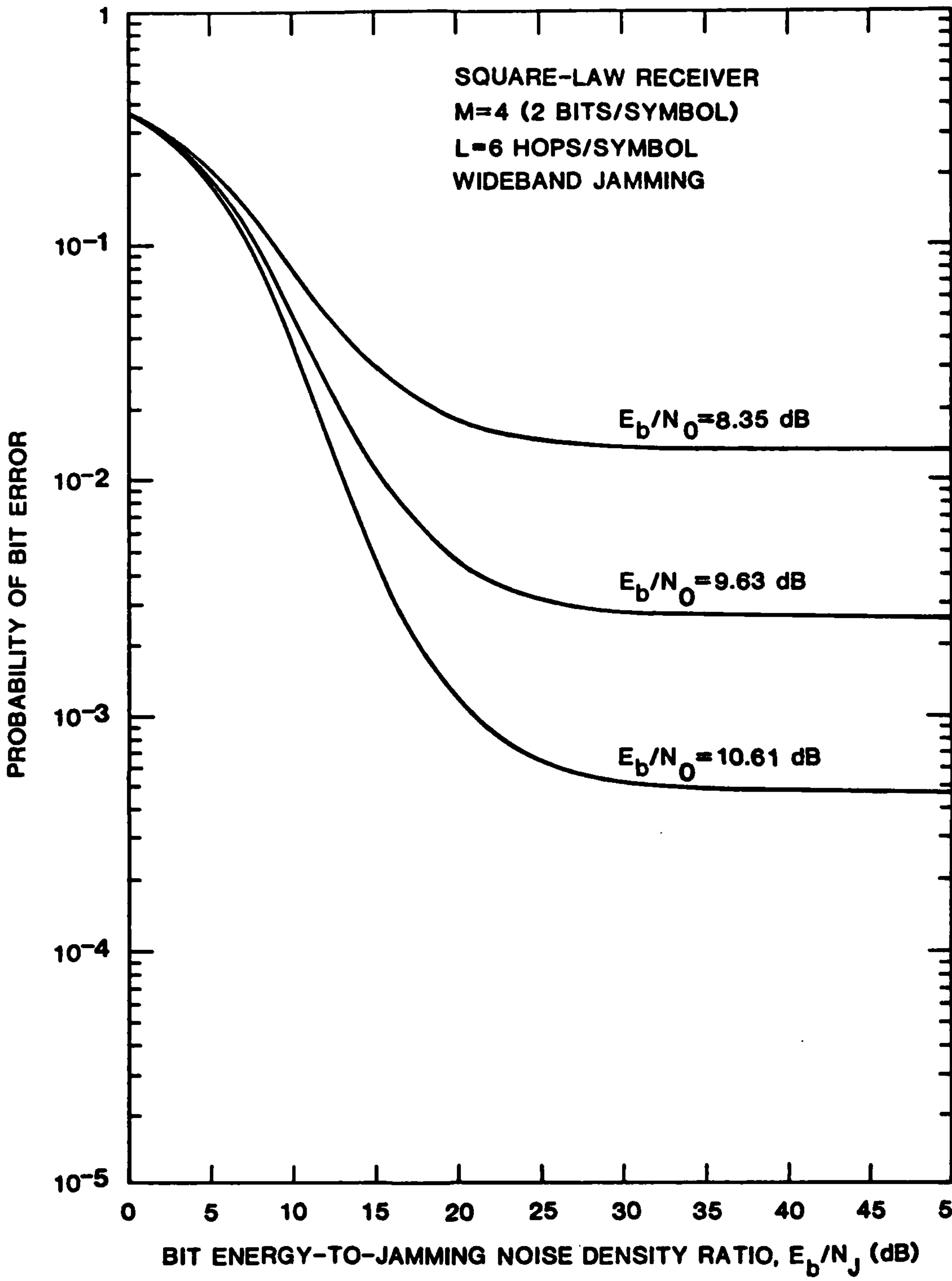


FIGURE 2-12 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK (M=4) SQUARE-LAW RECEIVER FOR L=6 HOPS/SYMBOL WITH SIGNAL-TO-THERMAL NOISE RATIO (E_b/N_0) AS A PARAMETER

performance due to noncoherent combining loss when L=4 and L=6 hops/bit. For example, when E_b/N_0 = 10.61 dB the error rate approaches asymptotes of 1.8×10^{-4} in Figure 2-11 and 4.8×10^{-4} in Figure 2-12, whereas in Figure 2-10 with L=1 this value of E_b/N_0 gives an asymptote of 1.0×10^{-5}. Thus the noncoherent combining loss increases as L increases and has degraded the bit error performance by more than an order of magnitude in the thermal-noise-limited region for L=4 and L=6 hops/symbol.

Figures 2-13 through 2-15 show the performance under wideband jamming of the square-law linear receiver for 8-ary FSK/FH with L=1, 4, and 6 hops/symbol, respectively, with E_b/N_0 as a parameter. As for the case of 4-ary FSK/FH, we see that the thermal-noise-limited region begins around E_b/N_J = 30 dB. Comparison of Figure 2-14 with Figure 2-11 and Figure 2-15 with Figure 2-12 shows the dramatic impact of noncoherent combining loss in the thermal-noise-limited region.

Figures 2-16 through 2-18 show the performance under wideband jamming of the square-law linear combining receiver for 16-ary FSK/FH with L=1, 4, and 6 hops/symbol, respectively, with the signal-to-thermal noise ratio E_b/N_0 as a parameter. As for the 4-ary and 8-ary cases, we see that the thermal-noise-limited region begins around E_b/N_J = 30 dB. Comparison of these figures with those for M=4 and M-8 clearly shows the increasing impact of noncoherent combining loss in the thermal-noise-limited region as M increases.

Figures 2-19 through 2-21 show the bit error probability for 4-ary, 8-ary, and 16-ary FSK/FH, respectively, with E_b/N_0 chosen to give an error rate of 10^{-5} without jamming (for L=1 hop/symbol), with the number (L) of hops/symbol as a parameter. These curves clearly show that the degradation of performance due to noncoherent combining persists over the entire range of

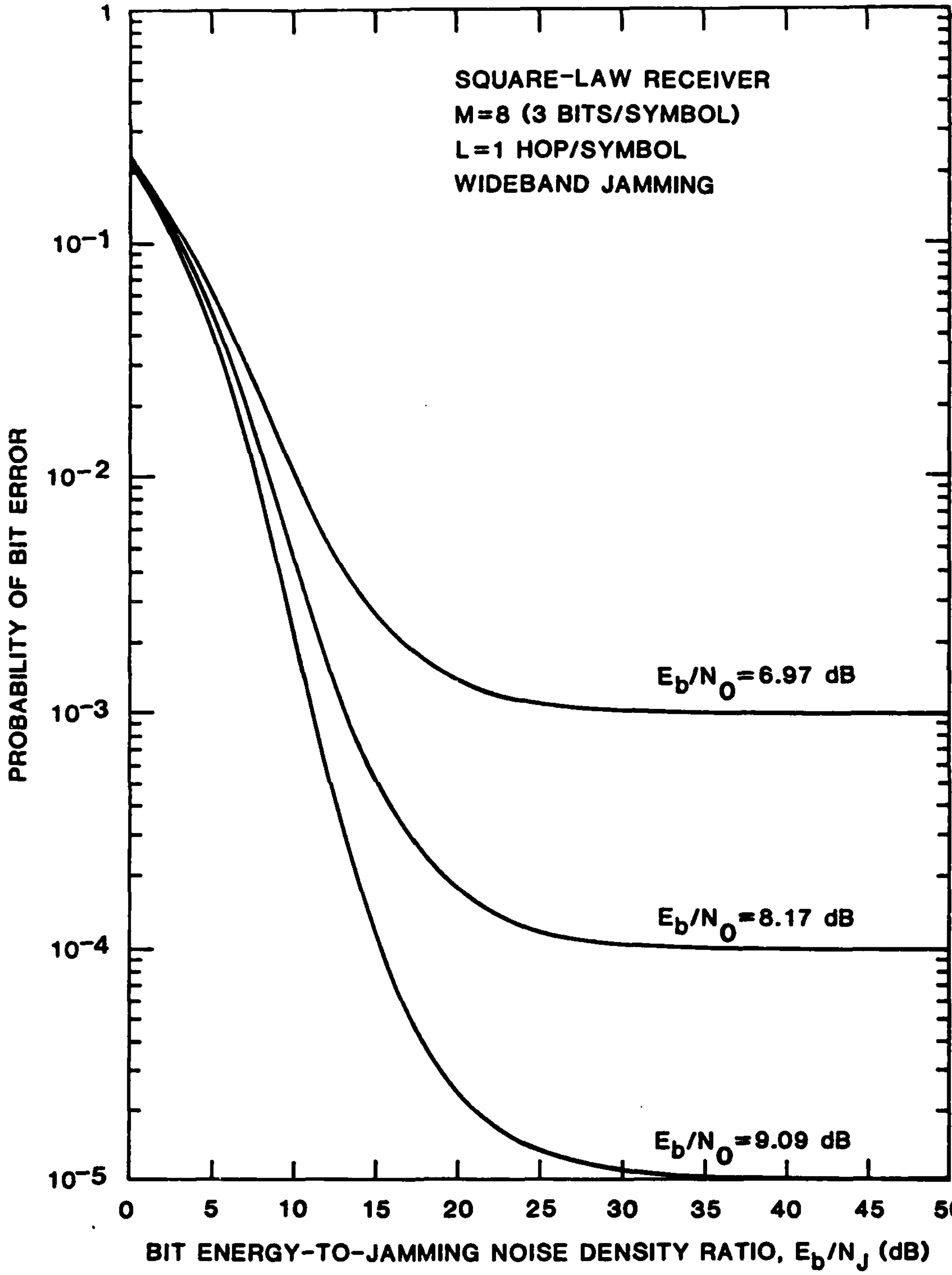


FIGURE 2-13 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK (M=8) SQUARE-LAW RECEIVER FOR L=1 HOP/SYMBOL WITH SIGNAL-TO-THERMAL NOISE RATIO (E_b/N_0) AS A PARAMETER

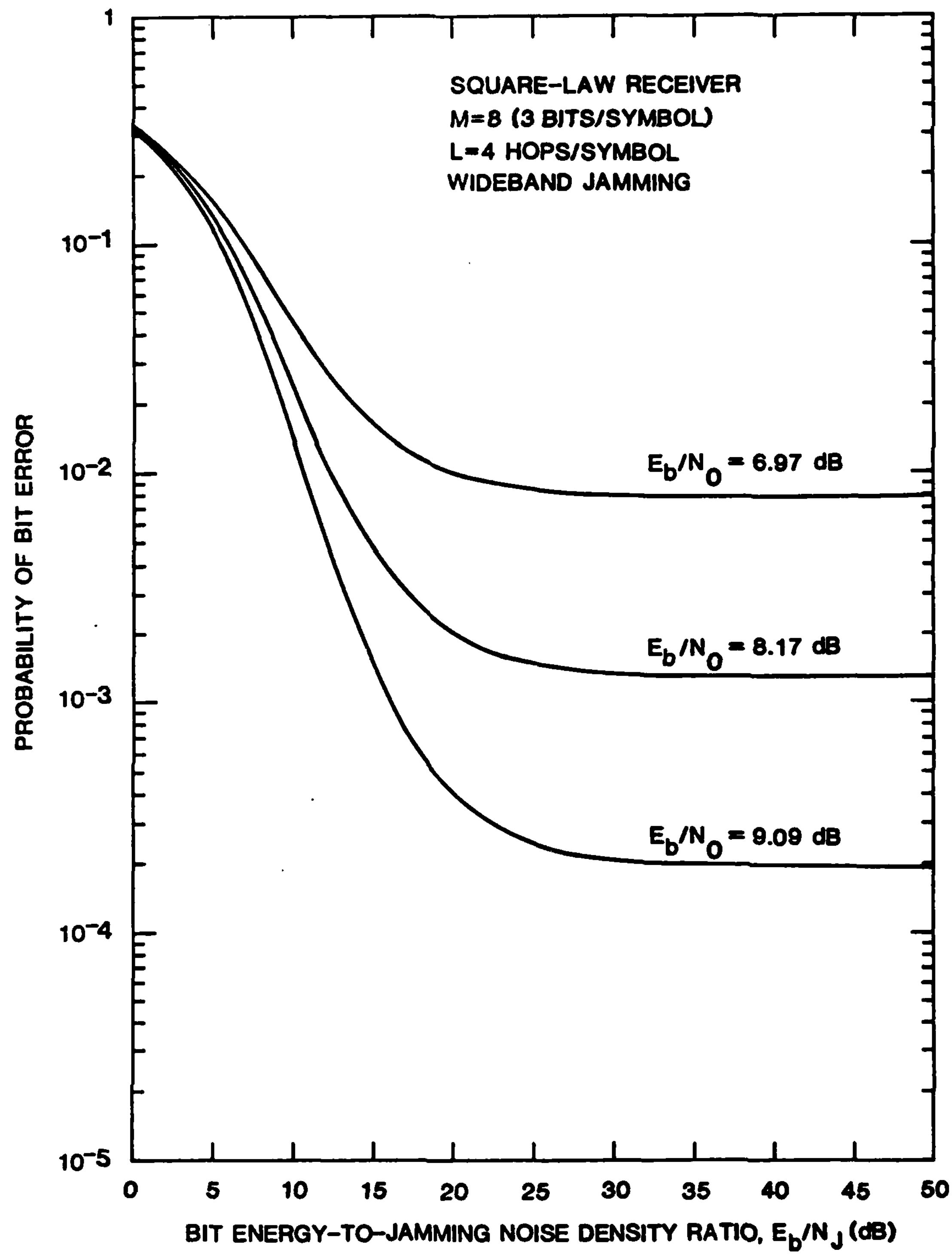


FIGURE 2-14 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK (M=8) SQUARE-LAW RECEIVER FOR L=4 HOPS/SYMBOL WITH SIGNAL-TO-THERMAL NOISE RATIO (E_b/N_0) AS A PARAMETER

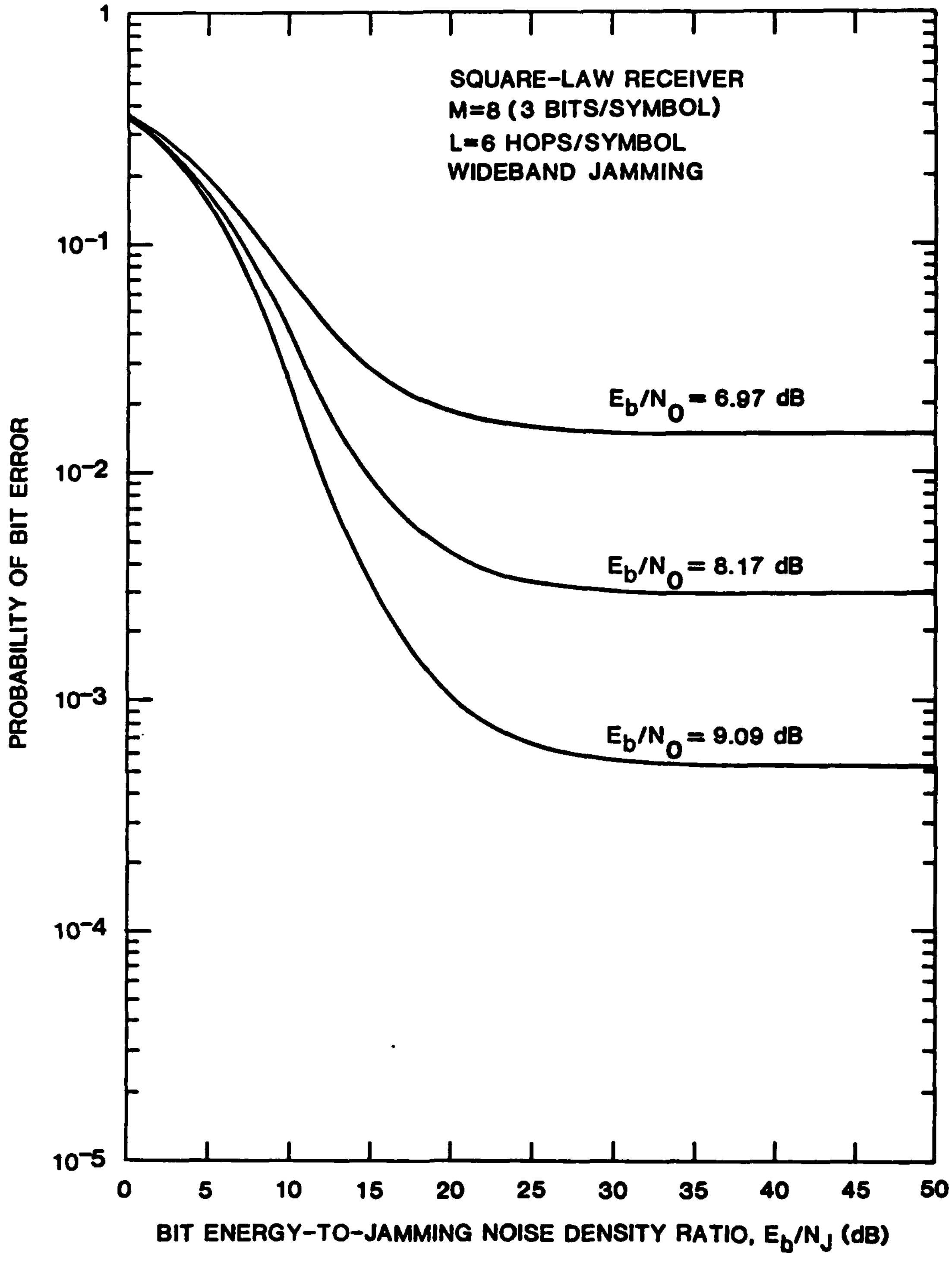


FIGURE 2-15 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK (M=8) SQUARE-LAW RECEIVER FOR L=6 HOPS/SYMBOL WITH SIGNAL-TO-THERMAL NOISE RATIO (E_b/N_0) AS A PARAMETER

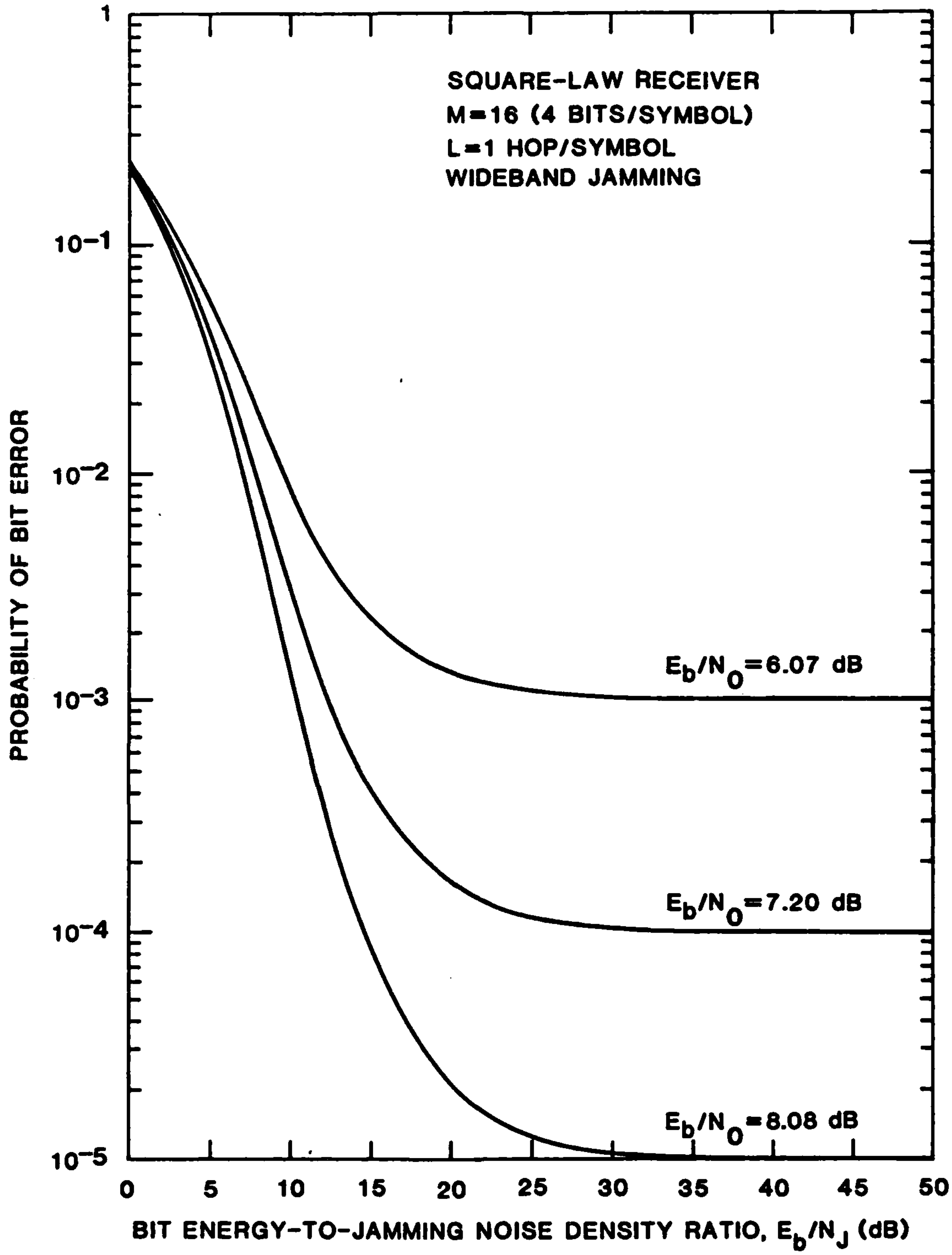


FIGURE 2-16 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK (M=16) SQUARE-LAW RECEIVER FOR L=1 HOP/SYMBOL WITH SIGNAL-TO-THERMAL NOISE RATIO (E_b/N_0) AS A PARAMETER

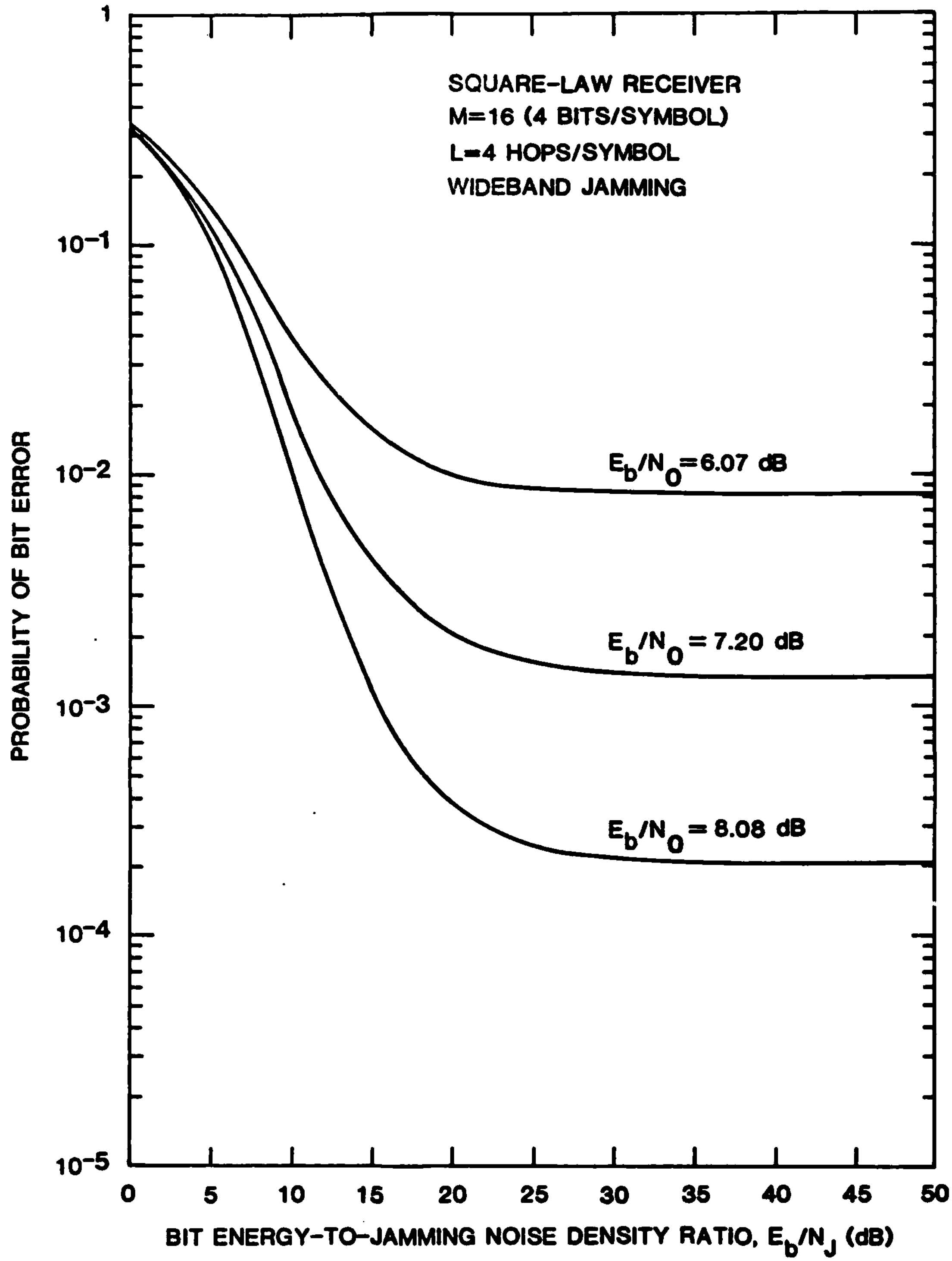


FIGURE 2-17 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK (M=16) SQUARE-LAW RECEIVER FOR L=4 HOPS/SYMBOL WITH SIGNAL-TO-THERMAL NOISE RATIO (E_b/N_0) AS A PARAMETER

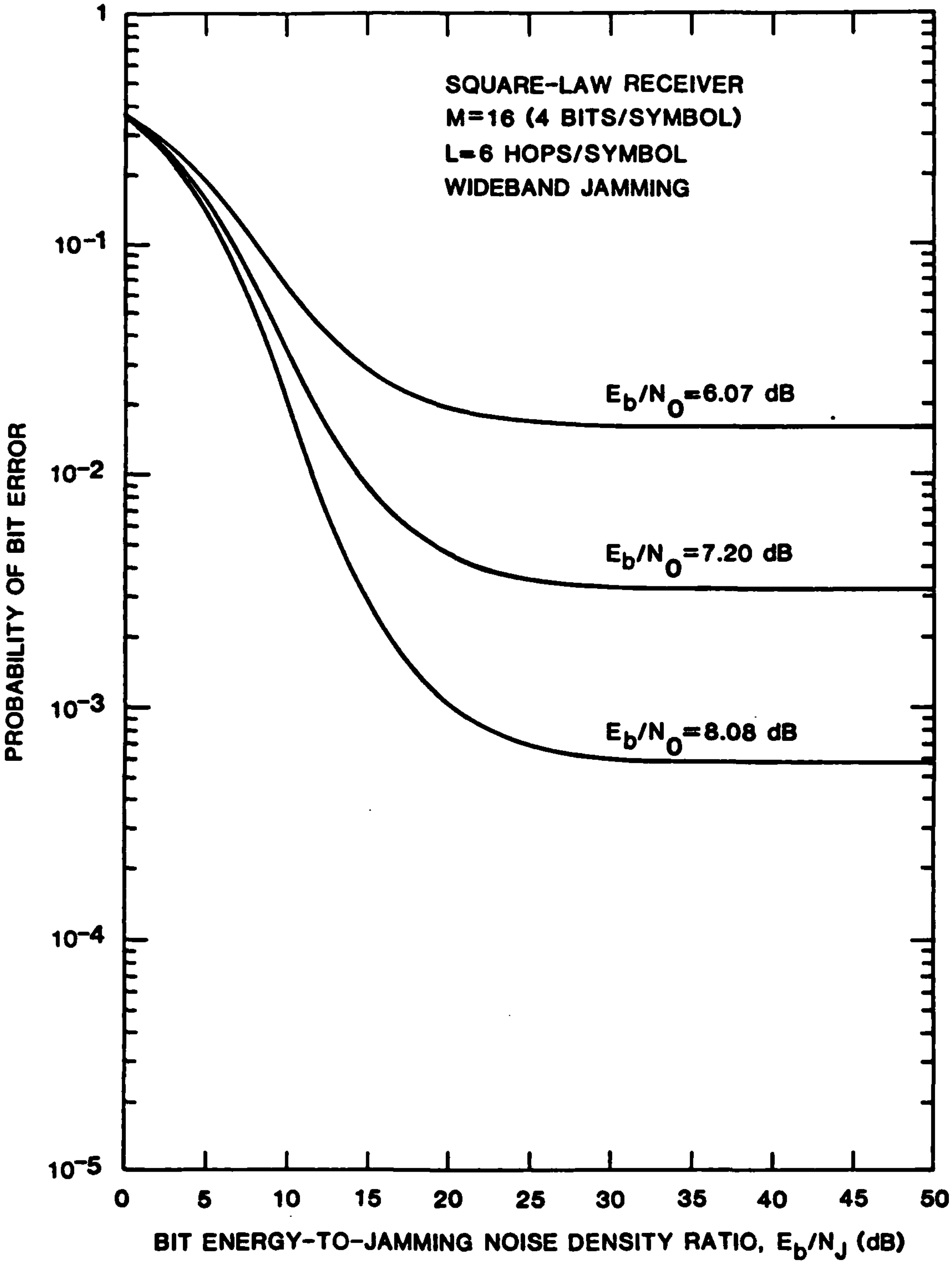


FIGURE 2-18 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK (M=16) SQUARE-LAW RECEIVER FOR L=6 HOPS/SYMBOL WITH SIGNAL-TO-THERMAL NOISE JAMMING RATIO (E_b/N_0) AS A PARAMETER

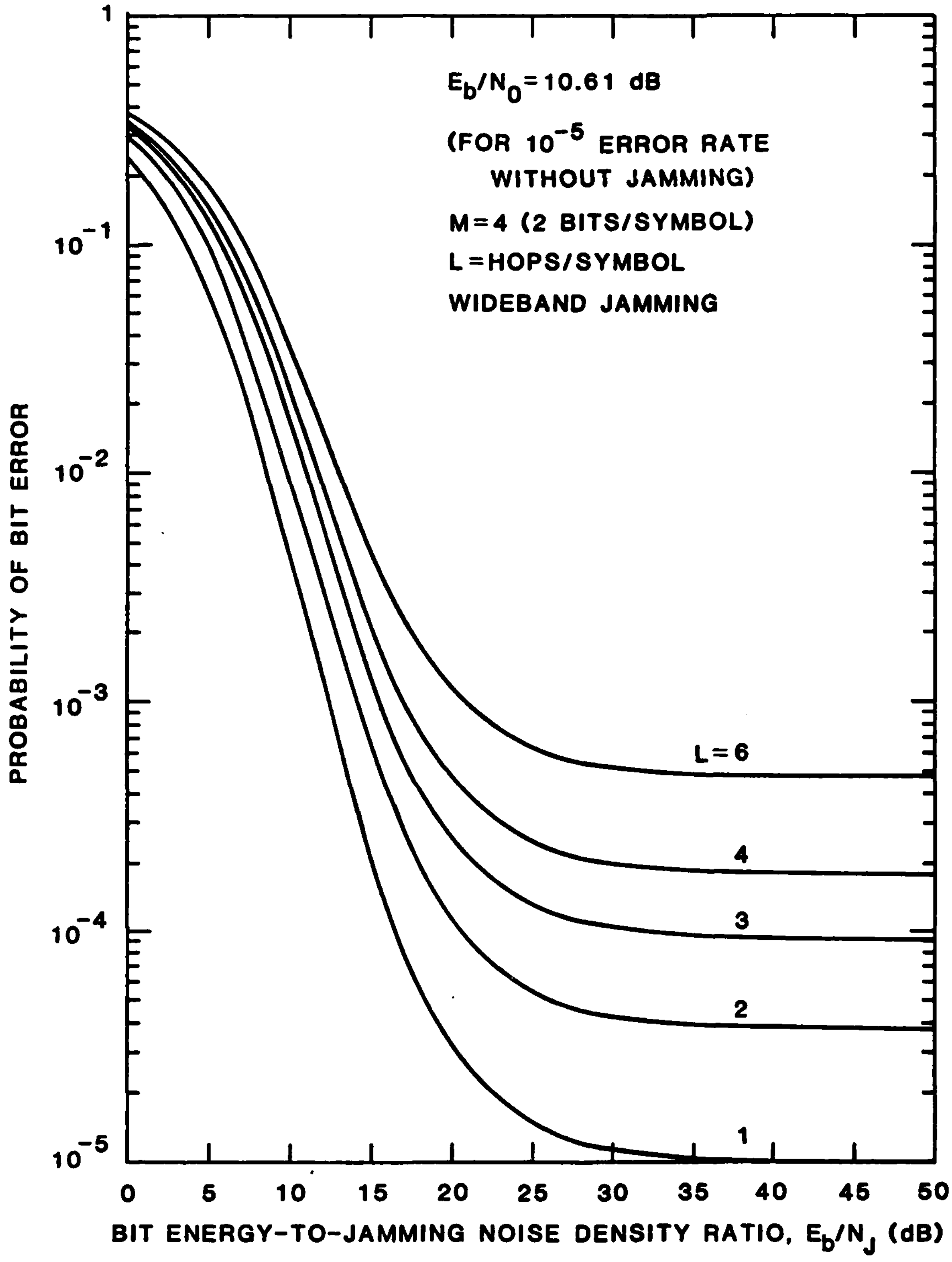


FIGURE 2-19 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK (M=4) SQUARE-LAW RECEIVER WHEN E_b/N_0 = 10.61 dB WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER

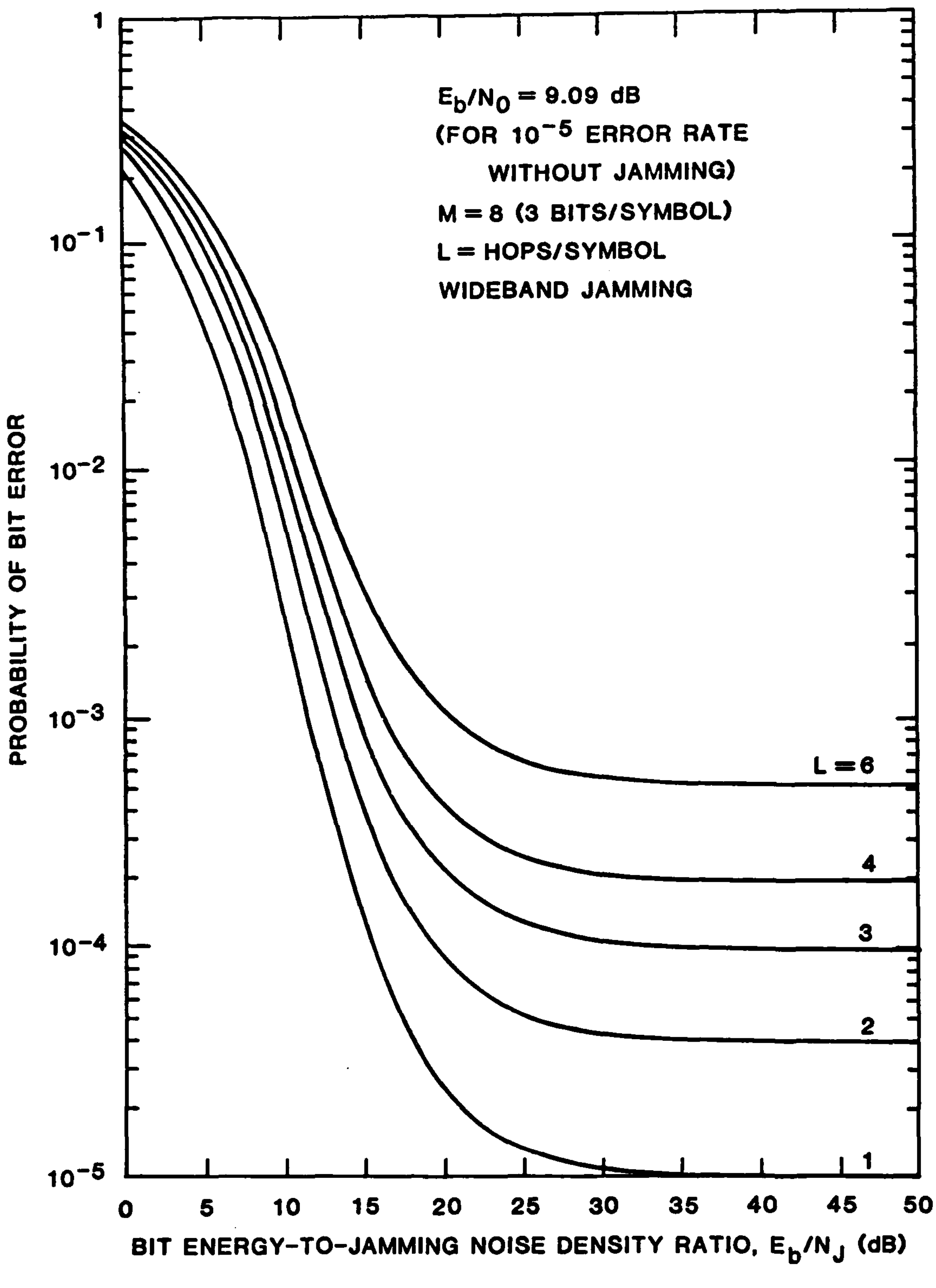


FIGURE 2-20 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK (M = 8) SQUARE-LAW RECEIVER WHEN E_b/N_0 = 9.09 dB WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER

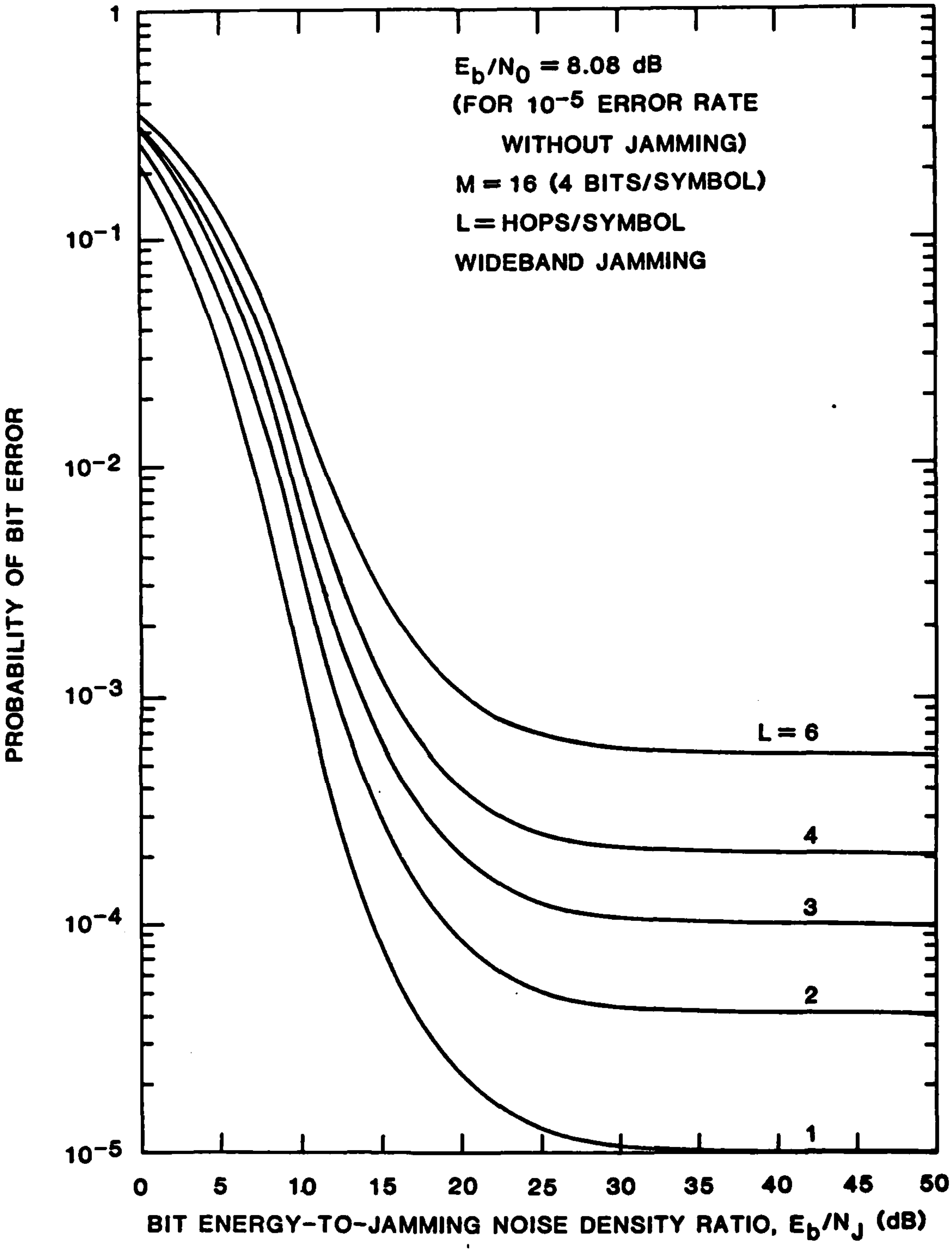


FIGURE 2-21 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK (M = 16) SQUARE-LAW RECEIVER WHEN E_b/N_0 = 8.08 dB WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER

E_b/N_J. Thus we conclude that there is no "diversity improvement" with the square-law linear combining receiver under wideband jamming.

Finally, Figures 2-22 through 2-24 show the bit error probability for L=1, 4, and 6 hops/symbol, respectively, with M as a parameter. In each case the choice of E_b/N_0 is coupled with M such that $P_e = 10^{-5}$ in the absence of jamming (for L=1). For one hop per symbol, Figure 2-22, we see that increasing M gives uniformly better performance for all E_b/N_J. But for L=4 and L=6 hops/symbol, Figures 2-23 and 2-24 show that the increase of combining loss with increasing M results in crossovers of the curves. When jamming is significantly strong, the M-ary coding gives a performance improvement in bit error rate; but in the thermal-noise-limited region the increase of combining loss with increasing M dominates and the binary (M=2) system gives the best performance. Comparsion of Figures 2-22 through 2-24 shows that the M-ary coding gain is nearly constant at a fixed $P_b(e)$ as L increases. For example, at $P_b(e) = 10^{-2}$, all three curves show a gain of 4 dB for M=32 relative to the curve for M=2.

2.4.2 Numerical Results for Partial-Band Jamming

The equations derived in Section 2.3, by themselves, are too complicated to give an immediate insight into the performance of an FH/MFSK system using a square-law linear combining receiver. Graphical examples of the performance curves are much more readily comprehended. Therefore, in this section we give a selected set of graphical results.

The numerical computation of bit error probability for the square-law linear combining receiver is a difficult task. Our first attempt was made using (2-87). Two problems arose in these numerical computations: the numerical computations were excessively slow and floating-point overflows occurred in the sequence of computations. The slowness arises from the very

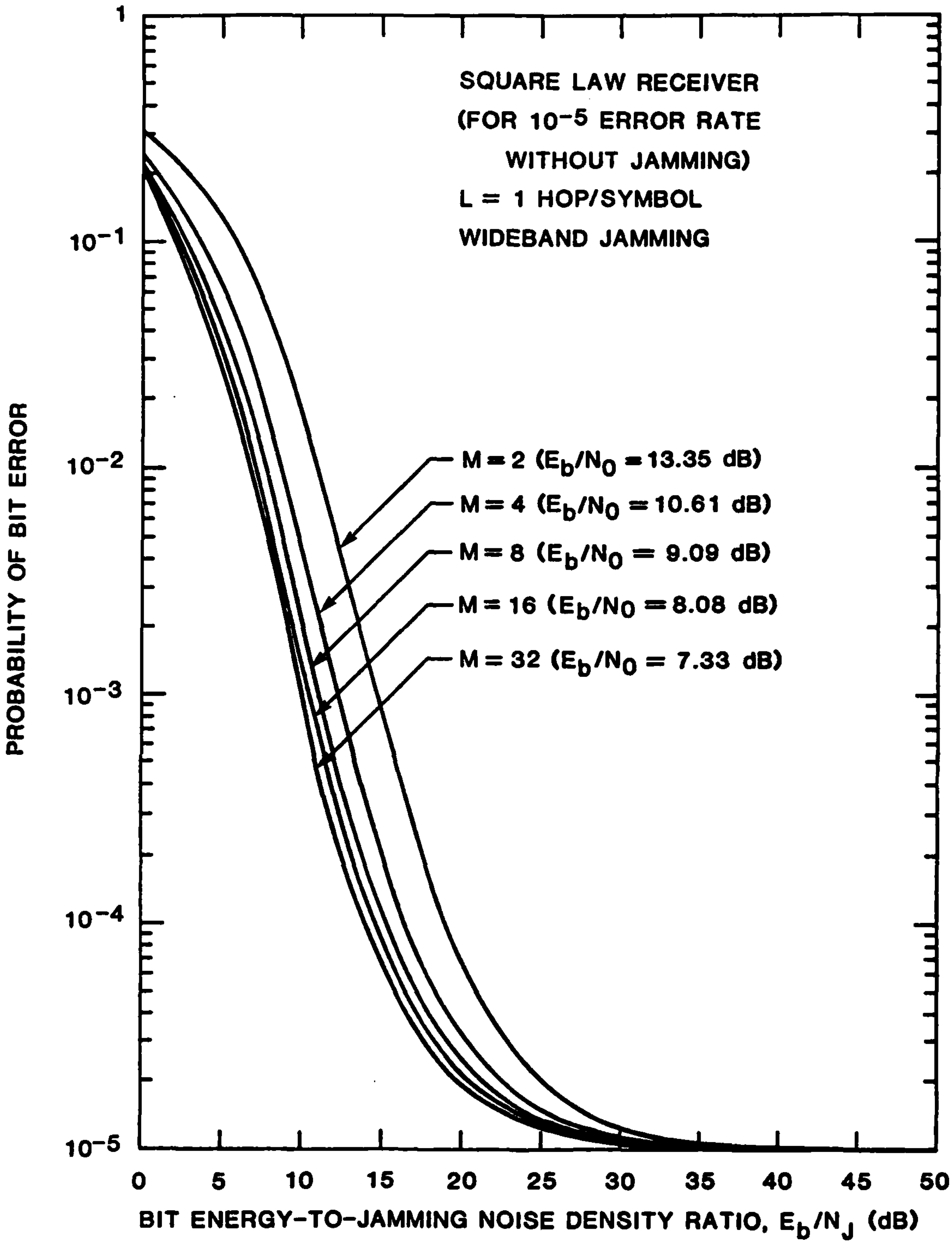


FIGURE 2-22 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK SQUARE-LAW RECEIVER FOR L=1 HOP/SYMBOL WHEN THE ASYMPTOTIC BIT ERROR RATE (L=1) = 10^{-5} WITH THE ALPHABET SIZE (M) AS A PARAMETER

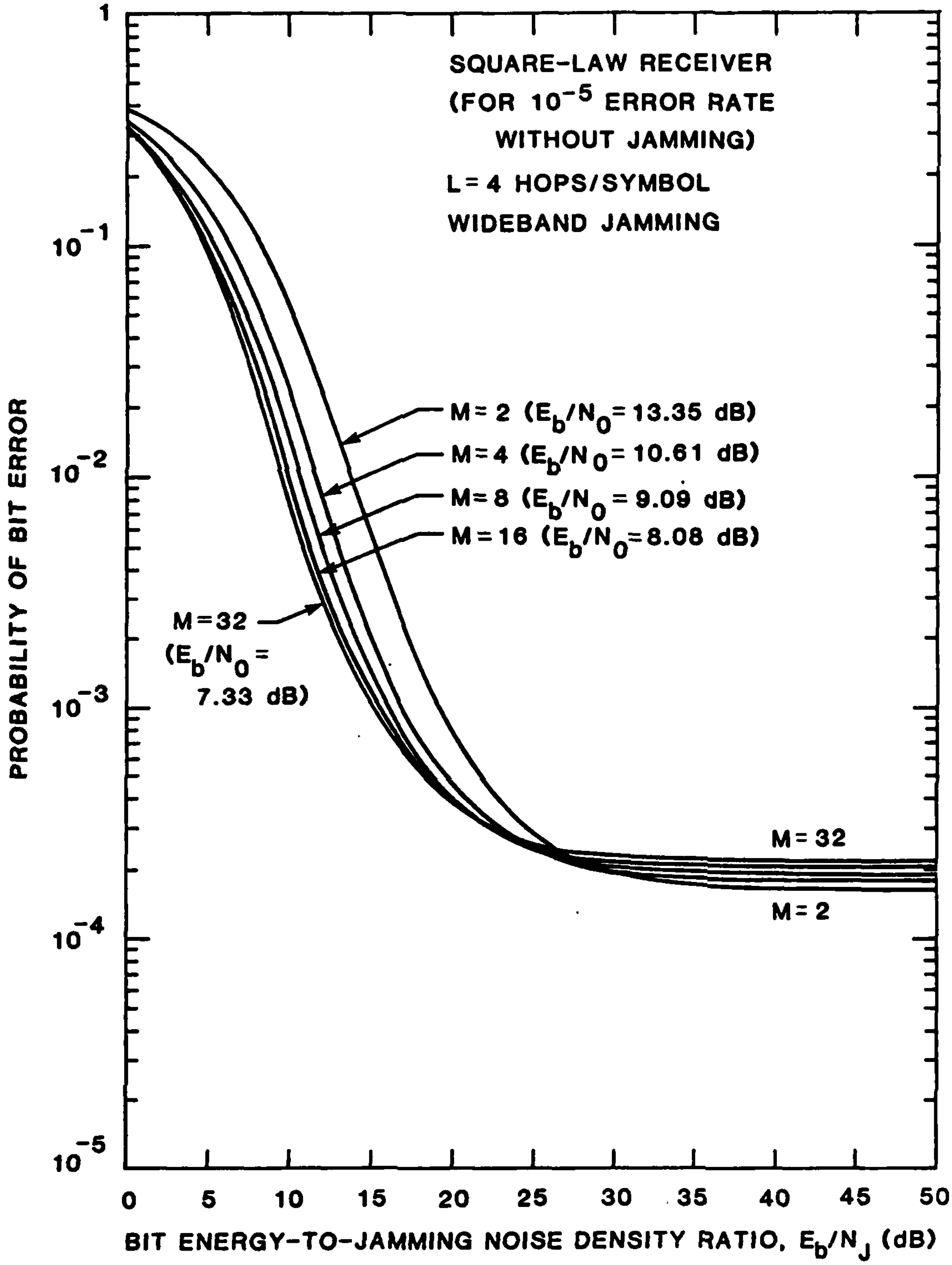


FIGURE 2-23 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK SQUARE-LAW RECEIVER FOR L = 4 HOPS/SYMBOL WHEN THE ASYMPTOTIC BIT ERROR RATE (L = 1) = 10^{-5} WITH THE ALPHABET SIZE (M) AS A PARAMETER

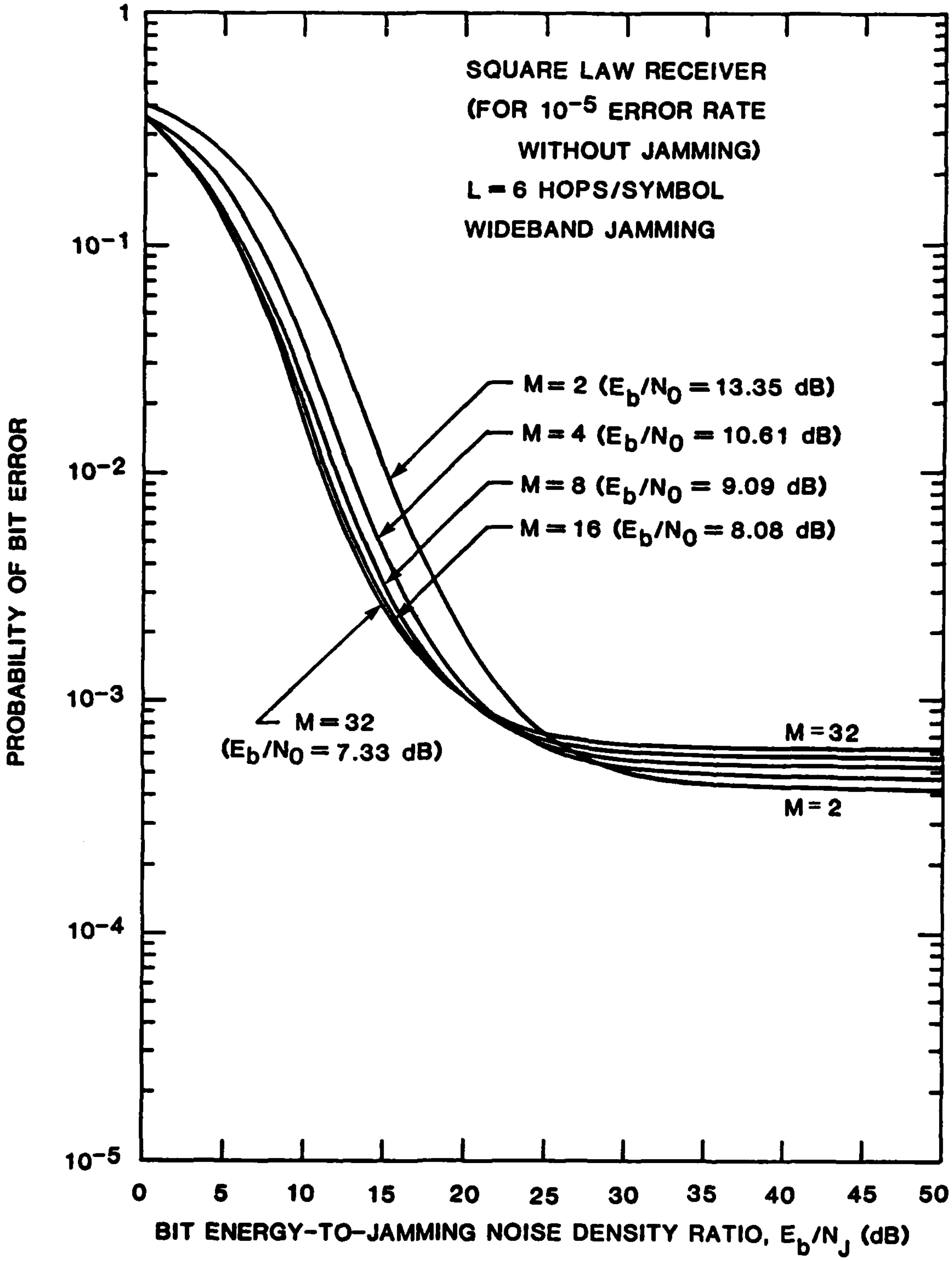


FIGURE 2-24 WIDEBAND JAMMING PERFORMANCE OF FH/MFSK SQUARE-LAW RECEIVER FOR L = 6 HOPS/SYMBOL WHEN THE ASYMPTOTIC BIT ERROR RATE (L = 1) = 10^{-5} WITH THE ALPHABET SIZE (M) AS A PARAMETER

complicated function which must be numerically integrated, compounded by the double summation (with infinite limits) of such integrals. The floating-point overflows occur as a result of the factors $(\ell\rho_T)^m/m!$ and $[(L-\ell)\rho_N]^n/n!$ which become quite large for moderate values of ρ_N or ρ_T. The point at which floating-point overflow occurs on the PDP-11/44 computer used for the computations is approximately 10^{38}, which is reached by a factor of the form $x^n/n!$ for $x > 91$. A third problem also arises from round-off errors due to the alternating signs in several of the summations.

A second approach involved direct numerical computation of the double integral in (2-61) using the program in Appendix 2G. Although problems with overflows and underflows remained for some parameter ranges, a few numerical results were obtained and are presented in Table 2-2. However, the computer time required for a single evaluation of (2-61) to 3- to 4-place accuracy was excessive, as shown by the column headed "Computer Time Used" in the table.

Therefore, we tried to compute the numerical results using (2-107). Again, numerical problems arose. The summation over the index n in (2-107) converges slowly because $(k+L)_n/n! \rightarrow 1$ as $n \rightarrow \infty$; several hundred terms are required to evaluate this sum. However, the recursion relation (2-99) for the coefficients c_{nm} becomes unstable due to round-off errors. Using double precision floating point arithmetic on the PDP-11/44, the coefficients c_{nm} can be computed successfully using (2-99) only for n up to about 20. These difficulties were further compounded by underflows causing premature termination of the summation over the index k.

The only exception to the numerical difficulties outlined above was for the special case of L=1 hop/symbol. In this case, the troublesome terms vanish from the equations and results are readily obtained using the program given in Appendix 2H. Encouraged by this, we sought specialized equations,

TABLE 2-2

NUMERICAL RESULTS OBTAINED USING (2-61)

M	L	E_b/N_0	E_b/N_J	γ	$P_b(e)$	COMPUTER TIME USED *
4	2	13.35 dB	10 dB	1.0×10^{-3}	9.98×10^{-4}	0.693 hr.
4	2	13.35 dB	10 dB	1.0×10^{-2}	9.51×10^{-3}	
4	2	9.64 dB	10 dB	1.0×10^{-2}	4.12×10^{-2}	1.200 hr.
4	2	9.64 dB	10 dB	2.5×10^{-1}	3.39×10^{-2}	1.285 hr.

* Wall-clock time with computer operations dedicated to this program.

rather than general equations, for numerical computations. A specialized equation for L=2 was derived (see Appendix 2F) and used successfully for limited ranges of E_b/N_J for all values of γ and for higher values of E_b/N_J for sufficiently small γ and for $\gamma=1$ (when most terms of the equation vanish), using the program given in Appendix 2I.

The behavior of the error rate as a function of the fraction of the band jammed is shown in Figures 2-25 and 2-26 for M=2 and M=4, respectively, for L=2 hops/symbol and E_b/N_0 set to the value which is required for $P_b(e) = 10^{-5}$ for ideal MFSK (see Table 2-1). We see from these curves that there is an optimum value of γ, which we will denote by γ_0, for which $P_b(e)$ is maximized. This value is a function of M, L, E_b/N_J, and E_b/N_0.

Figures 2-27 and 2-28 show the effect of varying the alphabet size, M, for a fixed number of hops per symbol, L=1 and L=2, respectively. The curves are plotted for $\gamma = \gamma_0$ at each point. We observe a modest improvement in performance for a fixed E_b/N_J as M increases. Thus a small M-ary coding gain is achieved on the partial-band noise jamming channel.

In Figures 2-29 and 2-30, we show the effects of varying the number of hops/symbol for M=2 and M=4, respectively, under optimum partial-band jamming. In Figure 2-30 we have also plotted the wideband jamming results for comparison. We observe that for E_b/N_J on the order of 15 to 35 dB, the degradation of the communications link performance due to optimum partial-band jamming is an order of magnitude greater than the performance degradation due to wideband jamming. Hence the jammer must optimize the fraction γ in order to achieve maximum jamming effectiveness.

The conclusion which can be drawn from all of these figures is that the square-law linear combining receiver is not effective in combatting optimum partial-band noise jamming. Increasing the number of hops per symbol does not improve performance for this receiver structure.

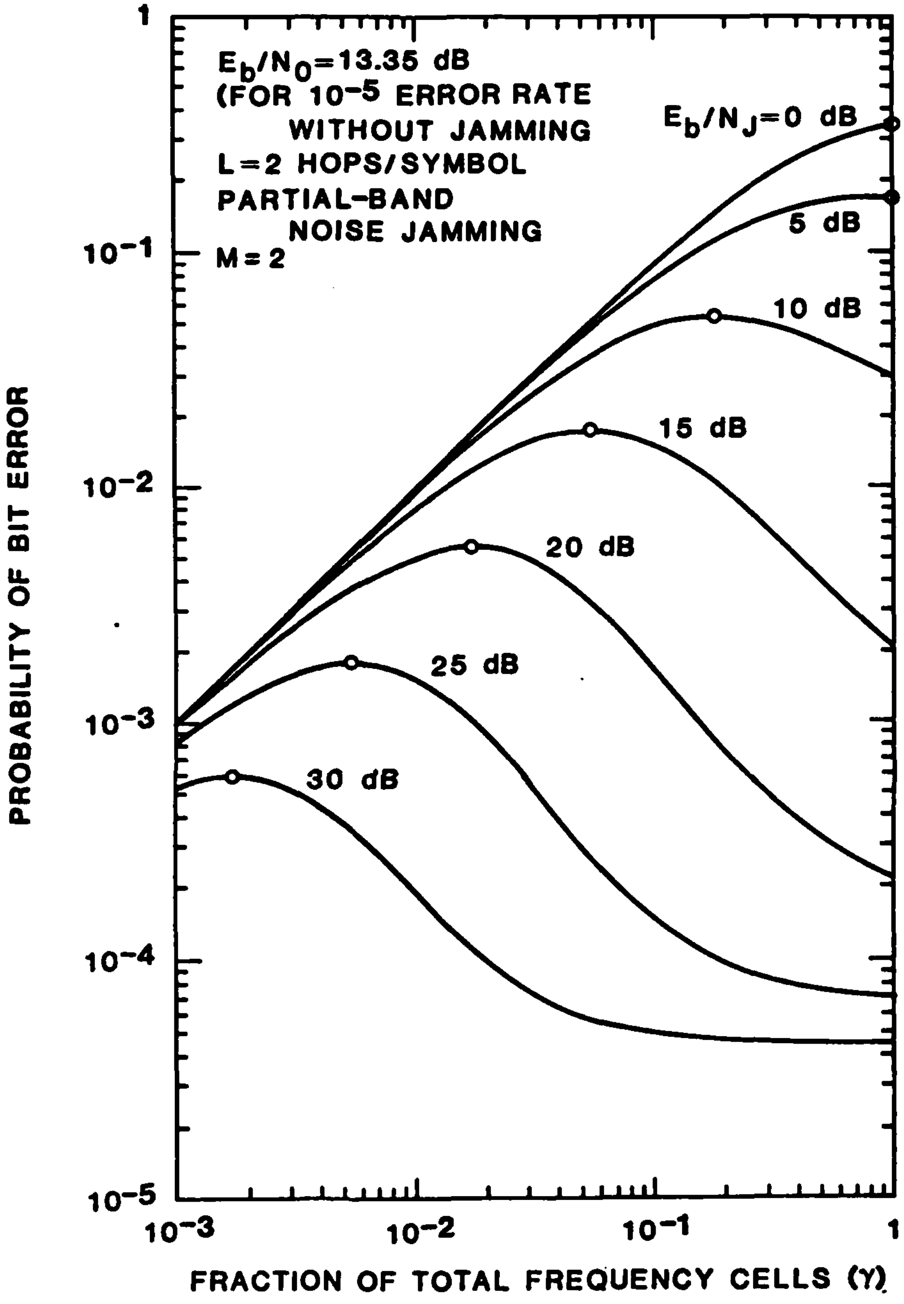


FIGURE 2-25 PROBABILITY OF ERROR VS. FRACTION OF TOTAL FREQUENCY CELLS JAMMED FOR L=2 HOPS/SYMBOL WHEN M=2 AND E_b/N_0=13.35 dB (E_b/N_J AS PARAMETER) WITH SQUARE-LAW LINEAR COMBINING RECEIVER

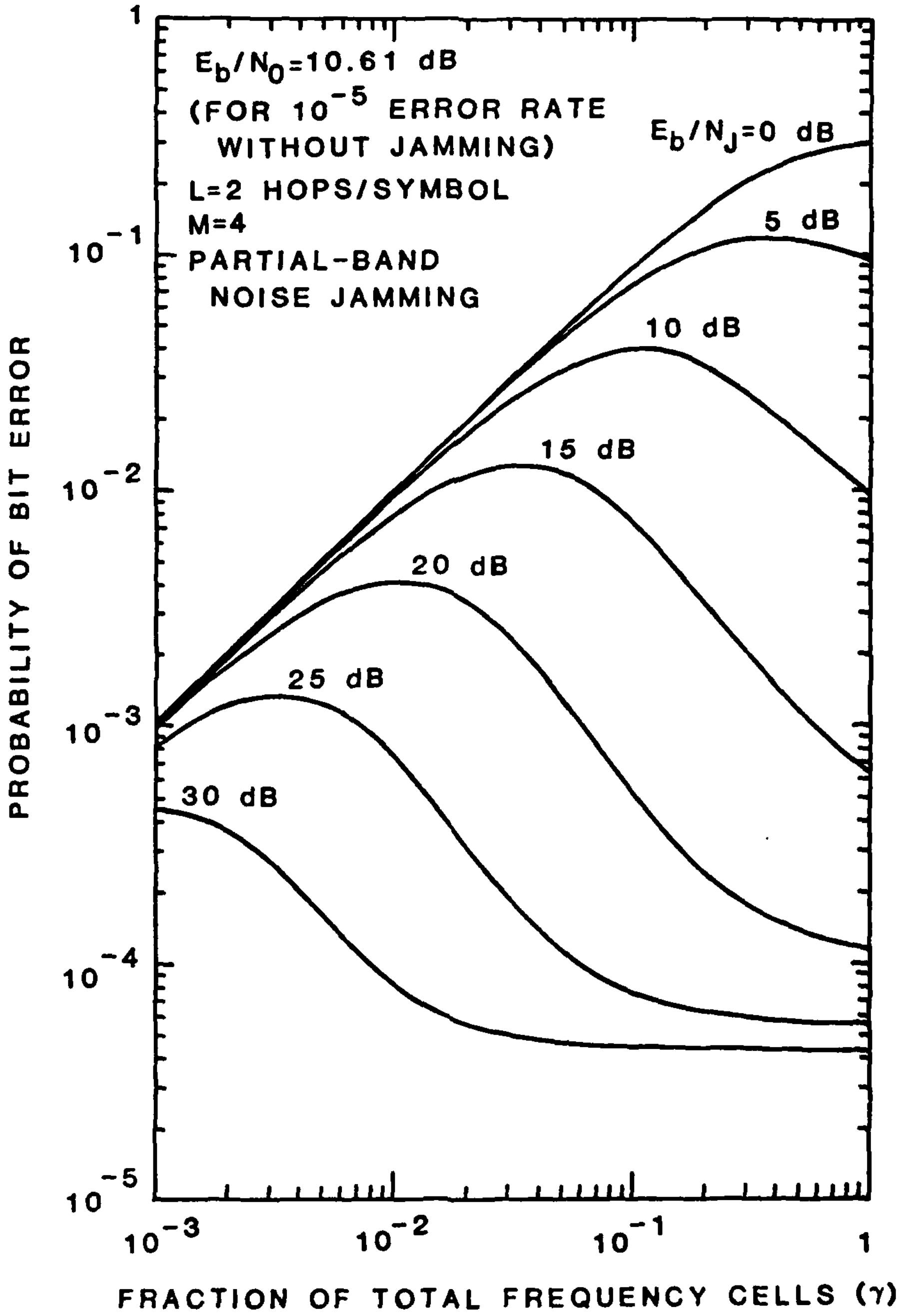


FIGURE 2-26 PROBABILITY OF ERROR VS. FRACTION OF TOTAL FREQUENCY CELLS JAMMED FOR L=2 HOPS/SYMBOL WHEN M=4 AND E_b/N_0=10.61 dB (E_b/N_J AS PARAMETER) WITH SQUARE-LAW LINEAR COMBINING RECEIVER

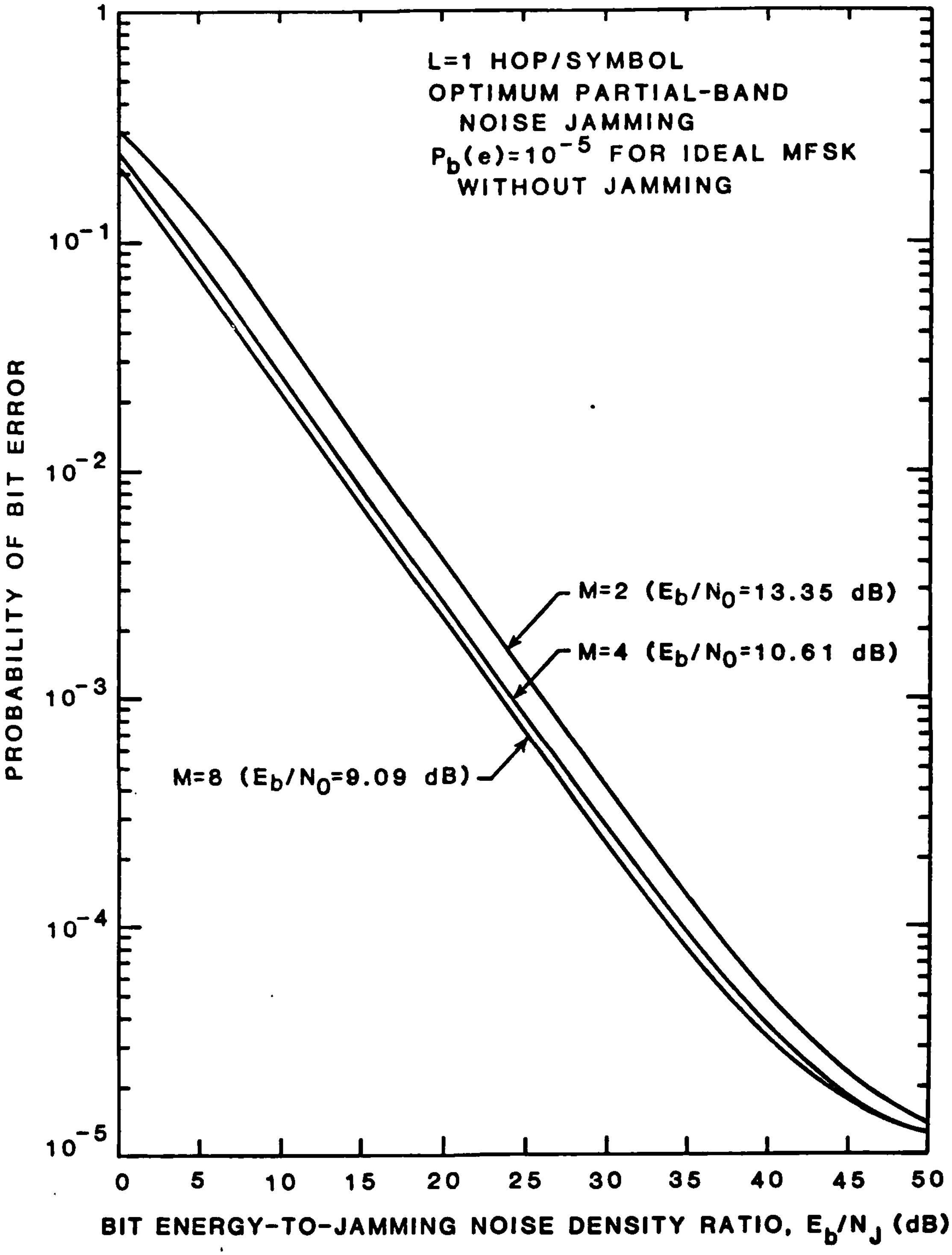


FIGURE 2-27 PROBABILITY OF ERROR VS. E_b/N_J WHEN L=1 AND E_b/N_0 IS SUCH THAT $P_b(e)=10^{-5}$ FOR IDEAL MFSK (M AS PARAMETER) WITH SQUARE-LAW LINEAR COMBINING RECEIVER

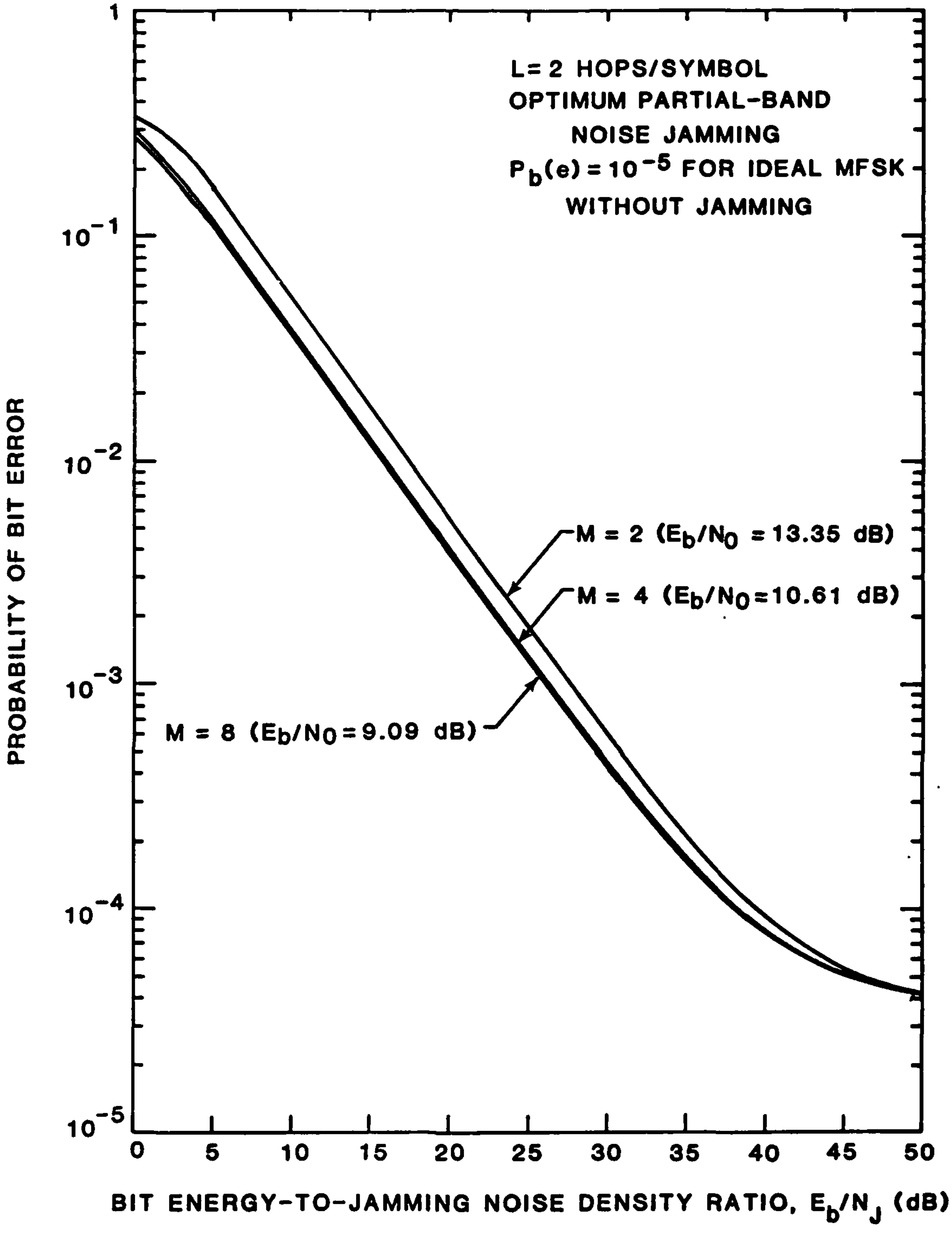


FIGURE 2-28 PROBABILITY OF ERROR VS. E_b/N_J WHEN L = 2 AND E_b/N_0 IS SUCH THAT $P_b(e) = 10^{-5}$ FOR IDEAL MFSK (M AS PARAMETER) WITH SQUARE-LAW LINEAR COMBINING RECEIVER

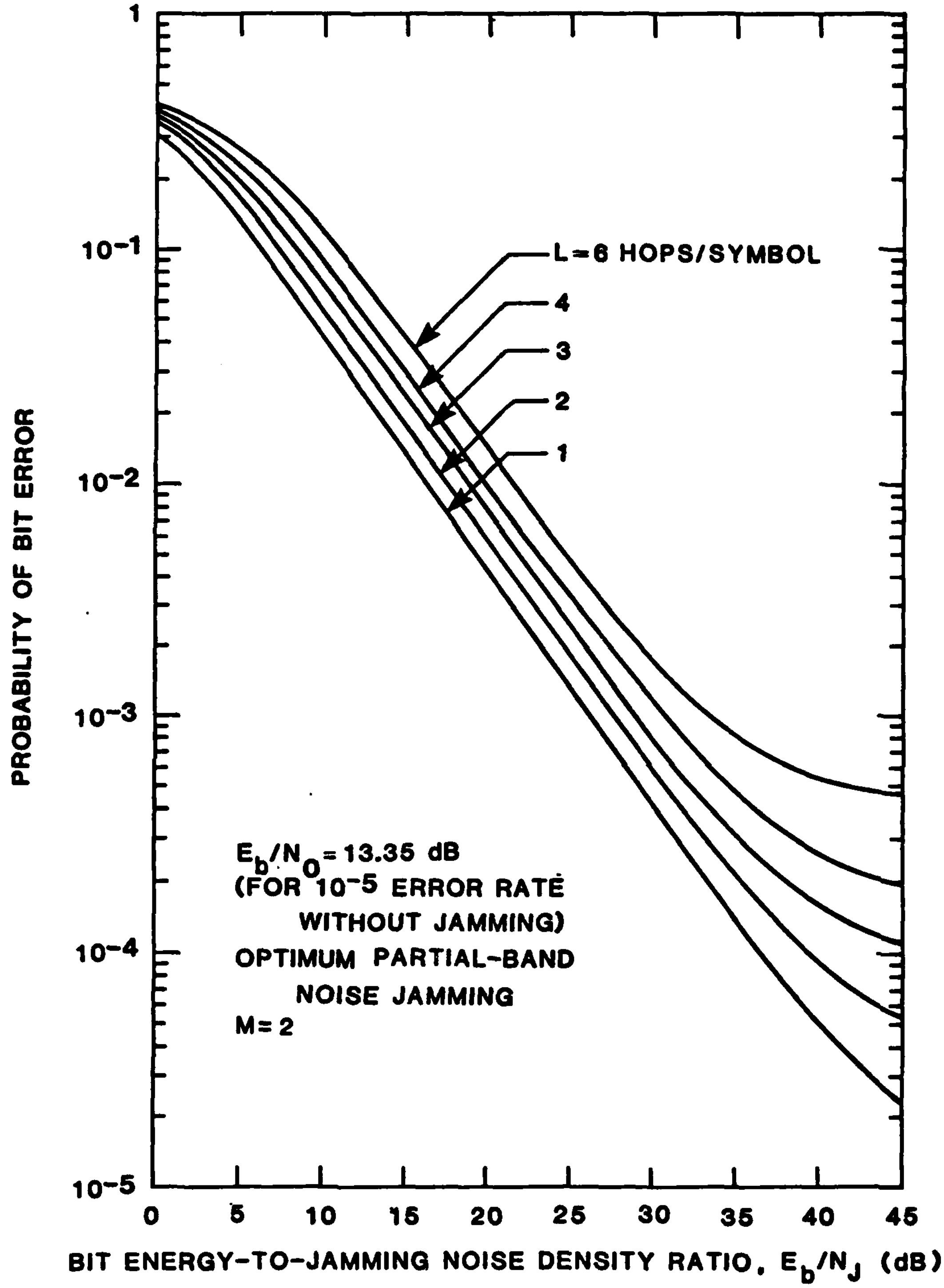


FIGURE 2-29 PROBABILITY OF ERROR VS. E_b/N_J WHEN M=2 AND E_b/N_0=13.35 dB (L AS PARAMETER) WITH SQUARE-LAW LINEAR COMBINING RECEIVER

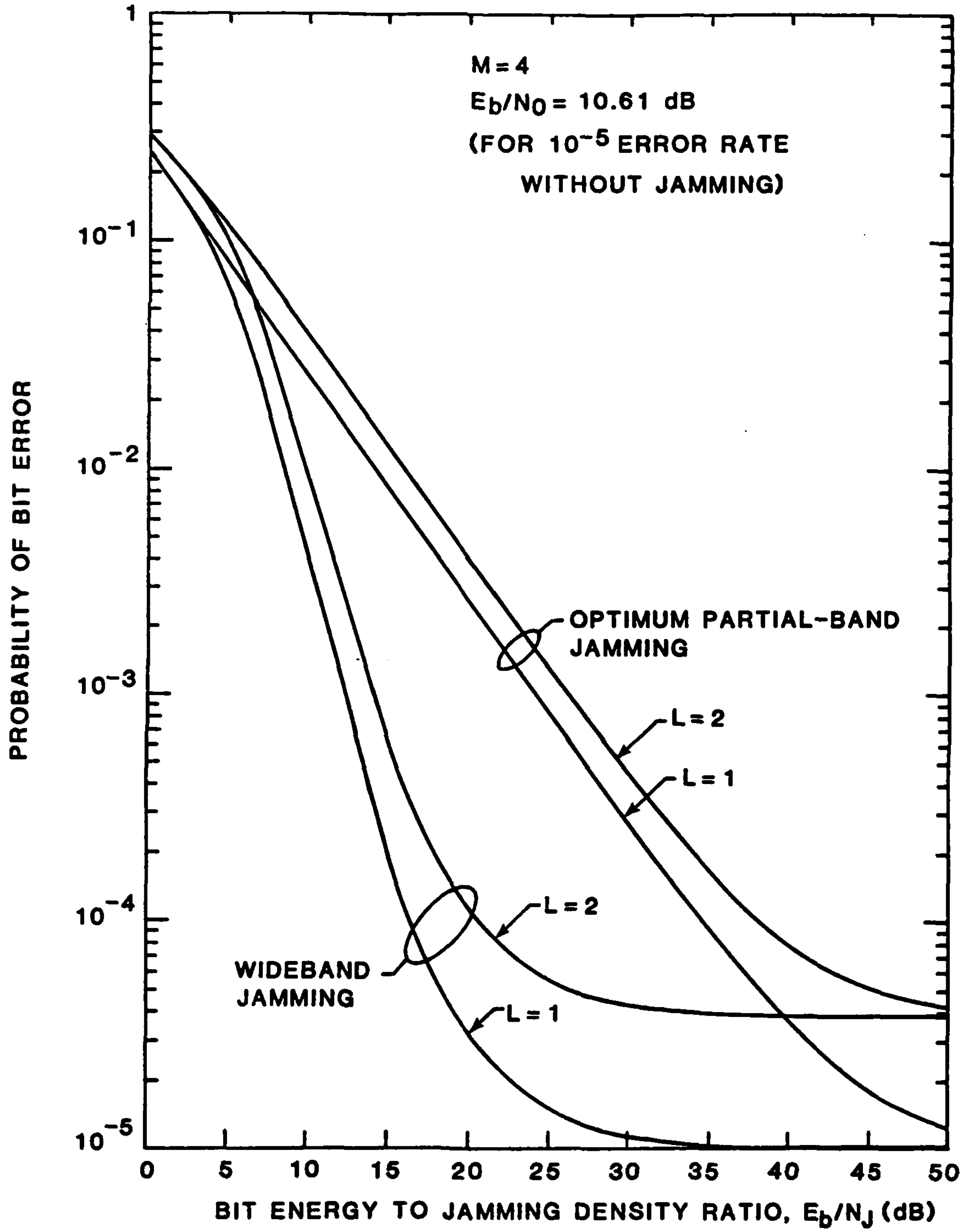


FIGURE 2-30 PROBABILITY OF ERROR VS. E_b/N_J WHEN $M = 4$ AND $E_b/N_0 = 10.61$ dB (L AS PARAMETER) WITH SQUARE-LAW LINEAR COMBINING RECEIVER

3.0 PERFORMANCE OF AN L-HOPS PER SYMBOL FH/MFSK RECEIVER WITH CLIPPERS UNDER PARTIAL-BAND NOISE JAMMING

Performance comparisons between the conventional square-law linear combining receiver and the square-law nonlinear combining receivers (i.e., clipper and AGC) for FH/BFSK waveforms (M=2) in the worst-case partial-band jamming channel without the simplifying assumption of neglecting the thermal noise [1] show that the conventional square-law linear combining receiver is the least effective when operated in a partial-band jamming channel as compared to the types of nonlinear combining receivers studied. This ranking also holds for the case of $M > 2$.

The purpose of this section is to present the exact analysis of the performance of the L-hops/symbol FH/MFSK nonlinear combining receiver with clippers for $M \geq 2$. The standard FH/MFSK receiver structure is modified by inserting clippers (soft limiters) prior to accumulating the envelope detector outputs.

The system we consider for the analysis is one in which the source produces one of a set of M equally likely symbols at time intervals of T_s seconds. The selected symbol from the source of rate $(\log_2 M)/T_s$ bits/second is transmitted by an L-hops/symbol transmission scheme; that is, each symbol which conveys $\log_2 M$ bits of information is broken into L independent transmissions each of duration T_s/L by means of frequency hopping over the system bandwidth W Hz. This frequency hopping takes place every τ seconds; the hopping rate is $1/\tau = L/T_s$ where τ is the hop dwell time.

The dehopped signal is assumed equally likely to be present in one of the M channels for the entire symbol period $T_s = L\tau$. The message signal decision $\hat{m}$ is taken to be index of the largest of the decision statistics z_i, $i = 1, 2, \ldots, M$.

The probability of error analysis for the L-hops/symbol FH/MFSK square-law combining receiver with clippers is described in subsection 3.2 and the error performance of the L-hops/symbol FH/MFSK linear-law combining receiver with clippers is described in subsection 3.3. A comparison of the performance of two receivers is given in subsection 3.4.

3.1 JAMMING MODEL

The total spread-spectrum system bandwidth is $W = NB$ Hz, where $B = 1/\tau$ is the cell bandwidth equal to the hopping rate and N is the number of channels (cells) available for hopping. A jamming power J watts is assumed to be distributed uniformly over a fractional bandwidth γW Hz, $0 < \gamma \leq 1$, so that the jamming power in the jammed cell of B Hz is given by

$$\sigma_J^2 = \left(\frac{J}{\gamma W}\right) B \text{ Watts}, \quad 0 < \gamma \leq 1. \tag{3-1}$$

Furthermore, we assume that the Gaussian thermal noise of uniform two-sided power spectral density $N_0/2$ W/Hz is also added to the signal at the receiver. We assume that the probability is γ that, on a given hop, all M of the hop frequency slots are jammed, and $1-\gamma$ that none are jammed. The effective spectral density of the jamming noise is taken to be $N_J/2\gamma$ when a hop is jammed, where $N_J \triangleq J/W$. Since the thermal noise n(t) and the jamming noise j(t) are additive Gaussian noises, the resultant noise power at the inputs to the envelope detectors may be written:

$$\sigma^2 = \begin{cases} \sigma_N^2 = NB & \text{with probability } 1-\gamma \\ \sigma_T^2 = \sigma_N^2 + \sigma_J^2 = (N_0 + N_J/\gamma)B & \text{with probability } \gamma. \end{cases} \tag{3-2}$$

3.2 ANALYSIS OF SQUARE-LAW COMBINING RECEIVER WITH CLIPPERS

The square-law combining receiver with clippers shown in Figure 3-1 uses a clipper (soft-limiter) in each of the M channels with fixed threshold η. The outputs of the clippers are then accumulated to provide the decision statistics for the M-ary decision. The clipping threshold is chosen to achieve the minimum error probability in the absence of jamming at a specified signal-to-thermal noise ratio (E_b/N_0). Note that if the clippers are removed, the resultant conventional structure is a near-optimum receiver for the Gaussian channel.

Without loss of generality, we assume that the signal with power S is in channel 1, or

$$s(t) = \sqrt{2S}\cos(\omega_1 t + \theta_k),\ (k-1)\tau \le t \le k\tau,\quad k = 1, 2, \ldots, L. \tag{3-3}$$

The combination of jamming and thermal noise on the kth hop produces the detector output samples

$$x_{1k} = \left(\sqrt{2S}\cos\theta_k + n_{c1k} + j_{c1k}\right)^2 + \left(\sqrt{2S}\sin\theta_k + n_{s1k} + j_{s1k}\right)^2 \tag{3-4a}$$

$$x_{ik} = \left(n_{cik} + j_{cik}\right)^2 + \left(n_{sik} + j_{sik}\right)^2,\quad i = 2, 3, \ldots, M \tag{3-4b}$$

where n_{cik}, n_{sik}, $i = 1, 2, \ldots, M$; $k = 1, 2, \ldots, L$, are the independent thermal noise quadrature components in the channels at the sample times $t_k = k\tau$ with

$$E\left\{n_{cik}^2\right\} = E\left\{n_{sik}^2\right\} = \sigma_N^2 = N_0 B \text{ for all } i, k, \tag{3-5}$$

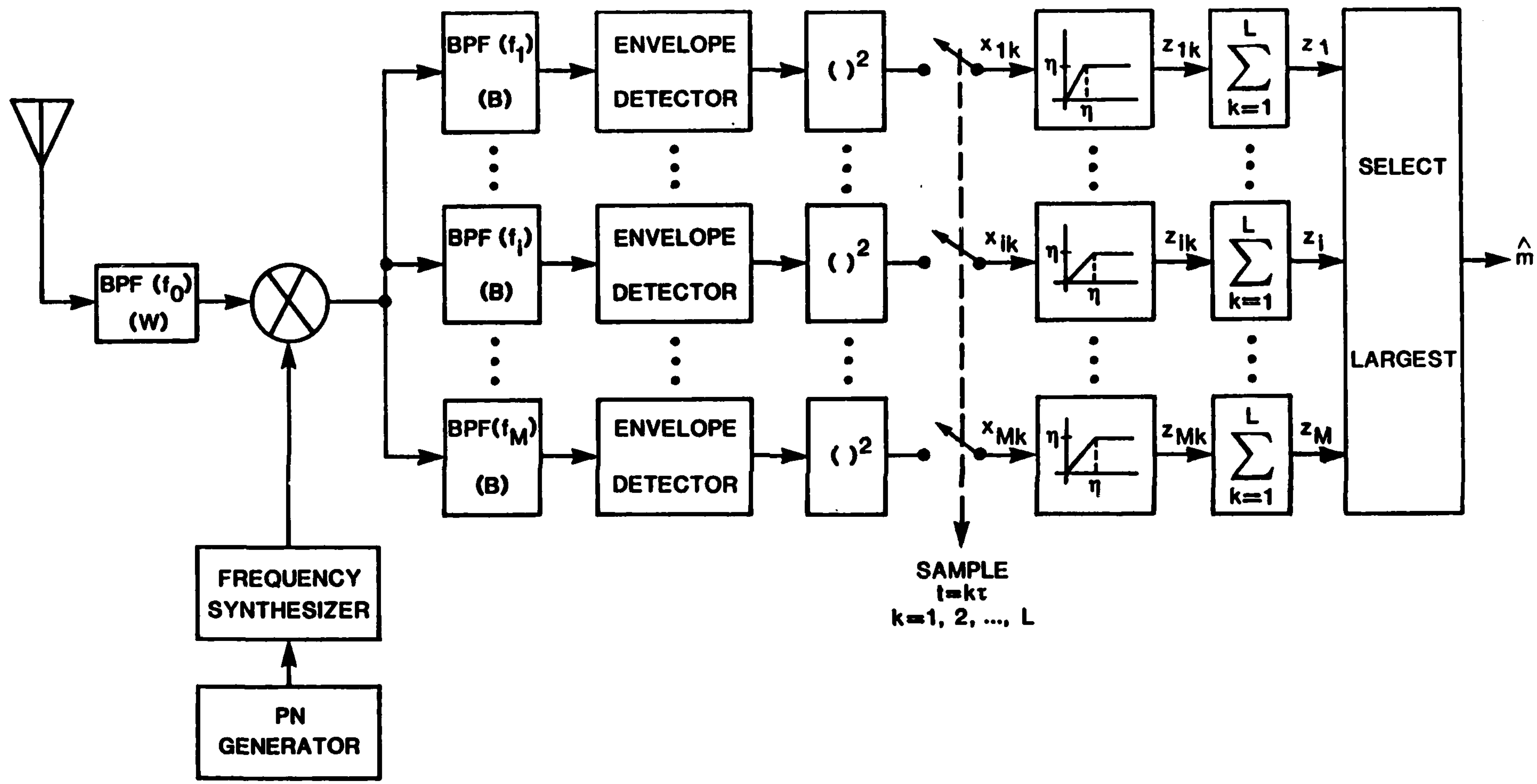


FIGURE 3-1 FH/MFSK SQUARE-LAW COMBINING RECEIVER WITH SOFT LIMITERS (CLIPPERS)

and j_{cik}, j_{sik}, $i = 1, 2, \ldots, M$; $k = 1, 2, \ldots, L$, are the independent jamming noise quadrature components in the channels at the sample times $t_k = k\tau$ with

$$E\left\{j_{cik}^2\right\} = E\left\{j_{sik}^2\right\} = \sigma_J^2 = N_J B/\gamma, \text{ for all } i, k. \tag{3-6}$$

The resultant noise power at the inputs to the envelope detectors is given by (3-2).

In order to analyze the clipper receiver, we found it convenient to use the uniformly quantized version of the soft-limiter characteristic shown in Figure 3-2. It was found [1] that for $L > 1$ the optimum clipping threshold is practically constant for $N=32$ and higher, where N is the number of quantization levels, indicating the results for $N=32$ are very close to those for the unquantized soft limiter. The quantization gives rise to the discrete-valued clipper outputs z_{ik}, $i = 1, 2, \ldots, M$; $k = 1, 2, \ldots, L$, with discrete probabilities given by

$$U_{in}\left(\sigma_k^2; S_i\right) = \Pr\left\{z_{ik} = \frac{n}{N-1}\eta\right\} = \Pr\left\{\frac{n}{N-1}\eta \le x_{ik} \le \frac{n+1}{N-1}\eta\right\}. \tag{3-7a}$$

Since $\sqrt{x_{ik}}$ are Rician ($i=1$) and Rayleigh ($i > 1$) random variables, (3-7a) may be expressed as

$$U_{1n} = \begin{cases} Q\left(\sqrt{2\rho}, \sqrt{\frac{n\eta}{N-1}b}\right) - Q\left(\sqrt{2\rho}, \sqrt{\frac{(n+1)\eta b}{N-1}}\right), & n = 0, 1, \ldots, N-2 \\ Q\left(\sqrt{2\rho}, \sqrt{\eta b}\right), & n = N-1 \end{cases} \tag{3-7b}$$

and

$$U_{in} = \begin{cases} e^{-\frac{n\eta b}{2(N-1)}} - e^{-\frac{(n+1)\eta b}{2(N-1)}}, & n = 0, 1, \ldots, N-2 \\ e^{-\frac{\eta b}{2}}, & n = N-1;\ i = 2, \ldots, M \end{cases} \tag{3-7c}$$

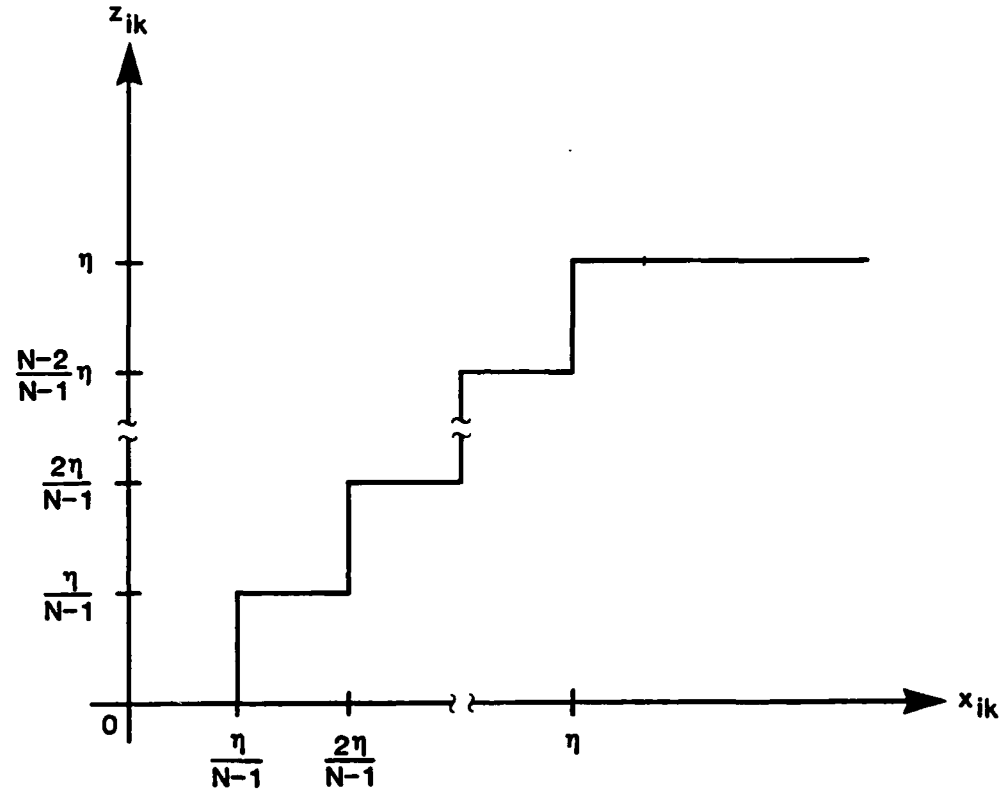


FIGURE 3-2 QUANTIZED SOFT LIMITER CHARACTERISTIC

where $Q(\alpha,\beta)$ is Marcum's Q-function and ρ is the signal-to-noise power ratio. The general expression in (3-7) can be written for the unjammed case by letting

$$\rho = \rho_N = \frac{S}{\sigma_N^2} \tag{3-7d}$$

and

$$b = \frac{1}{\sigma_N^2}; \tag{3-7e}$$

and for the jammed case by letting

$$\rho = \rho_T = \frac{S}{\sigma_T^2} = \frac{S}{\sigma_N^2 + \sigma_J^2} \tag{3-7f}$$

and

$$b = \frac{\gamma a}{(1+\gamma a)\sigma_N^2} = \frac{1}{\sigma_T^2} \tag{3-7g}$$

where γ is the jamming fraction and

$$a \triangleq \frac{E_b/N_J}{E_b/N_0} . \tag{3-7h}$$

We note that

$$\frac{S}{\sigma_N^2} = \frac{\log_2 M}{L} \cdot \frac{E_b}{N_0} . \tag{3-7i}$$

The output statistics z_i under this receiver model are also discrete-valued. Their probabilities,

$$U_{in}^{(L)} \left(\sigma_1^2, \sigma_2^2, \ldots, \sigma_L^2; S_i\right) = \Pr\left\{z_i = \frac{n}{N-1}\, \eta\right\}$$

$$n = 0, 1, \ldots, L(N-1), \tag{3-8}$$

are obtained numerically by discrete convolutions. These results are then used to compute the conditional symbol error probabilities, $P_s(e;\gamma|\ell)$.

The symbol error probability is the average over all jamming events of the conditional probability of error, given that ℓ out of L hops are jammed:

$$P_S(e;\gamma) = \sum_{\ell=0}^{L} \binom{L}{\ell} \gamma^{\ell} (1-\gamma)^{L-\ell} P_S(e;\gamma|\ell). \tag{3-9}$$

The conditional probability of error can be expressed as

$$P_S(e;\gamma|\ell) = 1 - \Pr\{\text{correct symbol decision; } \gamma|\ell\}. \tag{3-10a}$$

Assuming a signal in the f_1 channel,

$$\begin{aligned} \Pr\{\text{correct};\gamma|\ell\} &= \Pr\left\{z_1 = \max_i z_i\right\} \\ &+ \frac{1}{2}\Pr\left\{z_1 \text{ is one of two equal largest } z_i\right\} \\ &+ \frac{1}{3}\Pr\left\{z_1 \text{ is one of three equal largest } z_i\right\} \\ &+ \ldots + \frac{1}{M}\Pr\left\{\text{all } M\ z_i \text{ are equal}\right\}. \end{aligned} \tag{3-10b}$$

In the second and following terms in (3-10b), we assume that if two or more output statistics are equal, a randomized decision is made. After evaluating the probabilities, the final expression for the correct symbol decision is

$$\Pr\left\{\text{correct};\gamma|\ell\right\}$$

$$= \sum_{m=1}^{M} \frac{1}{m}\binom{M-1}{m-1} \Pr\left\{z_1 = z_2 = \ldots = z_m > z_{m+1},\ldots,z_1 = z_2 = \ldots . z_m > z_M;\gamma|\ell\right\}$$

$$= \sum_{m=1}^{M-1} \frac{1}{m}\binom{M-1}{m-1} \sum_{n=1}^{L(N-1)} \Pr\left\{z_1 = \frac{n\eta}{N-1}\right\}\left[\Pr\left(z_2 = \frac{n\eta}{N-1}\right)\right]^{m-1}$$

$$\times \left[\sum_{r=0}^{n-1} \Pr\left\{z_{m+1} = \frac{r\eta}{N-1}\right\}\right]^{M-m}$$

$$+ \frac{1}{M}\sum_{n=0}^{L(N-1)} \Pr\left\{z_1 = \frac{n}{N-1}\right\}\left[\Pr\left(z_2 = \frac{n\eta}{N-1}\right)\right]^{M-1} . \qquad \text{(3-11a)}$$

This may also may be written as

$$\Pr\left\{\text{correct};\ \gamma|\ell\right\}$$

$$= \sum_{n=1}^{L(N-1)} U_{1n}^{(L)}(\underline{\sigma}^2;S) \sum_{m=0}^{M-2} \binom{M-1}{m} \frac{1}{m+1}\left[U_{2n}^{(L)}(\underline{\sigma}^2;0)\right]^m \left[\sum_{r=0}^{n-1} U_{m+2,r}^{(L)}(\underline{\sigma}^2;0)\right]^{M-m-1}$$

$$+ \frac{1}{M}\sum_{n=0}^{L(N-1)} U_{1n}^{(L)}(\underline{\sigma}^2;S)\left[U_{2n}^{(L)}(\underline{\sigma}^2;0)\right]^{M-1} , \qquad \text{(3-11b)}$$

where $\underline{\sigma}^2 = (\sigma_1^2, \sigma_2^2,\ldots,\sigma_L^2)$ with ℓ of the σ_k^2, $k = 1, 2,\ldots,L$, equal to $(N_0 + N_J/\gamma)B$ and $L-\ell$ of them equal to N_0B.

The symbol error probability expressions in (3-9) through (3-11) can be better understood by examining the special case of M=2, i.e. BFSK. When M=2, (3-10) and (3-11) reduce to

$$P_s(e;\gamma|\ell, M=2) = 1 - \Pr\{\text{correct};\gamma|\ell, M=2\}$$

$$= 1 - \sum_{n=1}^{L(N-1)} U_{1n}^{(L)}(\underline{\sigma}^2;S) \sum_{r=0}^{n-1} U_{2r}^{(L)}(\underline{\sigma}^2;0) - \frac{1}{2} \sum_{n=0}^{L(N-1)} U_{1n}^{(L)}(\underline{\sigma}^2;S)\, U_{2n}^{(L)}(\underline{\sigma}^2;0)$$

$$= \sum_{n=0}^{L(N-1)-1} U_{1n}^{(L)}(\underline{\sigma}^2;S) \sum_{r=n+1}^{L(N-1)} U_{2r}^{(L)}(\underline{\sigma}^2;0) + \frac{1}{2} \sum_{n=0}^{L(N-1)} U_{1n}^{(L)}(\underline{\sigma}^2;S)\, U_{2n}^{(L)}(\underline{\sigma}^2;0). \tag{3-12}$$

Equation (3-12) is identical to the conditional error probability expression for FH/BFSK receiver with clippers found previously [1, p. 287].

3.3 ANALYSIS OF LINEAR-LAW COMBINING RECEIVER WITH CLIPPERS

The analytical approach used in the linear-law receiver with clippers illustrated in Figure 3-3 resembles that of square-law receiver with clippers. Assuming, without loss of generality, that the signal with power S is in channel 1, we again have s(t) as given by (3-3). With linear envelope detectors, the combination of jamming and thermal noise on the kth hop produces the envelope detector output samples

$$x_{1k} = \sqrt{\left(\sqrt{2S}\cos\theta_k + n_{c1k} + j_{c1k}\right)^2 + \left(\sqrt{2S}\sin\theta_k + n_{s1k} + j_{s1k}\right)^2} \tag{3-13a}$$

$$x_{ik} = \sqrt{\left(n_{cik} + j_{cik}\right)^2 + \left(n_{sik} + j_{sik}\right)^2}\,, \quad i = 2, 3, \ldots, M, \tag{3-13b}$$

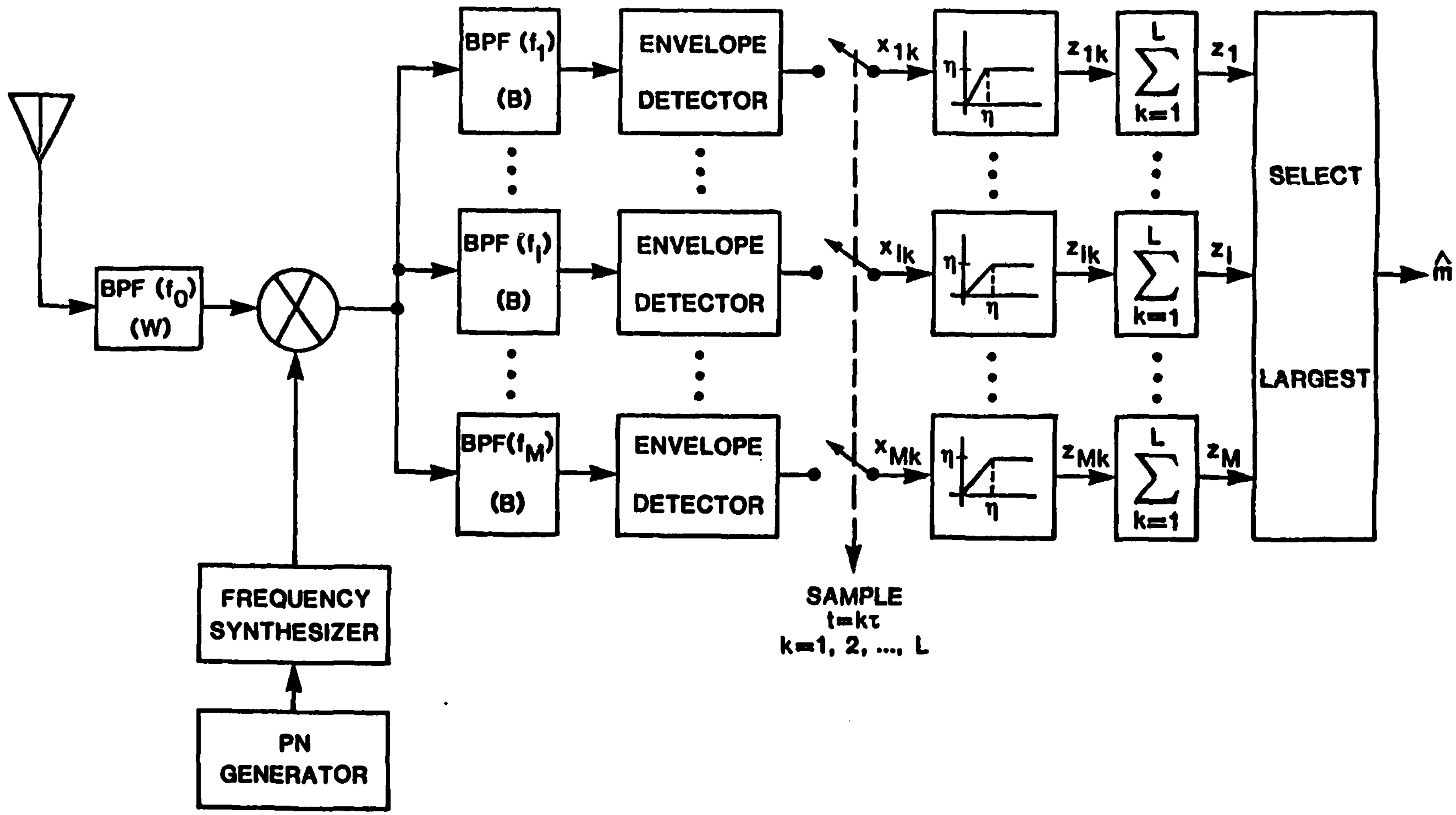


FIGURE 3-3 FH/MFSK LINEAR-LAW COMBINING RECEIVER WITH SOFT LIMITERS (CLIPPERS)

where n_{cik}, n_{sik}, and j_{cik}, j_{sik} are the independent quadrature components of the thermal noise and the jamming noise, respectively.

As shown in Figure 3-2, we use a quantized clipper (soft-limiter) with fixed threshold η in each of the M channels. The outputs of the clippers are accumulated to provide the decision statistics for the M-ary decision. The clipping threshold is chosen to achieve the minimum error probability in the absence of jamming.

The discrete-valued clipper outputs z_{ik}, $i = 1, 2, \ldots, M$; $k = 1, 2, \ldots, L$, for the linear-law receiver have the discrete probabilities

$$V_{in}\left(\sigma_k^2;\ S_i\right) = \Pr\left\{z_{ik} = \frac{n}{N-1}\eta\right\} = \Pr\left\{\frac{n}{N-1} \le x_{ik} < \frac{n+1}{N-1}\eta\right\}. \qquad \text{(3-14a)}$$

These probabilities may be evaluated as

$$V_{1n} = \begin{cases} Q\left(\sqrt{2\rho},\ \frac{n\eta b}{N-1}\right) - Q\left(\sqrt{2\rho},\ \frac{(n+1)\eta b}{N-1}\right), & n = 0, 1, \ldots, N-2 \\ Q\left(\sqrt{2\rho},\ \eta b\right), & n = N-1 \end{cases} \qquad \text{(3-14b)}$$

and

$$V_{in} = \begin{cases} e^{-\frac{1}{2}\left(\frac{n\eta b}{N-1}\right)^2} - e^{-\frac{1}{2}\left(\frac{(n+1)\eta b}{N-1}\right)^2}, & n = 0, 1, \ldots, N-2 \\ e^{-\frac{1}{2}(\eta b)^2}, & n = N-1, \end{cases}$$
$$i = 2, 3, \ldots M \qquad \text{(3-14c)}$$

where $Q(\alpha,\beta)$ is Marcum's Q-function and ρ is the signal-to-noise power ratio. The general expressions in (3-14b) and (3-14c) can be written for the unjammed case by letting

$$\rho = \rho_N = \frac{S}{\sigma_N^2} \tag{3-14d}$$

and

$$b = \frac{1}{\sigma_N}, \tag{3-14e}$$

and for the jammed case by letting

$$\rho = \rho_T = \frac{S}{\sigma_T^2} = \frac{S}{\sigma_N^2 + \sigma_J^2} \tag{3-14f}$$

and

$$b = \frac{1}{\sigma_N}\sqrt{\frac{a}{1+\gamma a}} = \frac{1}{\sigma_T} \tag{3-14g}$$

where γ is the jamming fraction and

$$a \triangleq \frac{E_b/N_J}{E_b/N_0}. \tag{3-14h}$$

The output decision statistics z_i under this receiver model are also discrete-valued; their probabilities

$$V_{in}^{(L)}\left(\sigma_1^2, \sigma_2^2, \ldots, \sigma_L^2; S_i\right) = \Pr\left\{z_i = \frac{n}{N-1}\eta\right\},$$

$$n = 0, 1, \ldots, L(n-1) \tag{3-15}$$

are obtained numerically by discrete convolutions. The symbol error probability is

$$P_s(e;\gamma) = \sum_{\ell=0}^{L} \binom{L}{\ell} \gamma^{\ell} (1-\gamma)^{L-\ell} P_s(e;\gamma|\ell). \tag{3-16}$$

Following the same analytical procedures defined previously in (3-10), the expression for the conditional error probability $P_s(e;\gamma|\ell)$ is

$$P_s(e;\gamma|\ell) = 1 - \Pr\{\text{correct};\ \gamma|\ell\}$$

$$= 1 - \sum_{n=1}^{L(N-1)} V_{1n}^{(L)}(\underline{\sigma}^2;S) \sum_{m=0}^{M-2} \binom{M-1}{m} \frac{1}{m+1} \left[V_{in}^{(L)}(\underline{\sigma}^2;0)\right]^m \left[\sum_{r=0}^{n-1} V_{in}^{(L)}(\underline{\sigma}^2;0)\right]^{M-m-1}$$

$$- \frac{1}{M} \sum_{n=0}^{L(N-1)} V_{1n}^{(L)}(\underline{\sigma}^2;S) \left[V_{in}^{(L)}(\underline{\sigma}^2;0)\right]^{M-1} \tag{3-17}$$

where $\underline{\sigma}^2 = (\sigma_1^2, \sigma_2^2, \ldots, \sigma_L^2)$ with ℓ of the σ_k^2 equal to $(N_0 + N_J/\gamma)B$ and $L-\ell$ of them equal to N_0B.

3.4 NUMERICAL RESULTS FOR CLIPPER RECEIVERS

The performances of the receivers with clippers are summarized graphically, both for wideband noise jamming ($\gamma=1$) and for optimum partial-band noise jamming. The performance in optimum or worst-case partial-band jamming was determined numerically by varying γ to find the maximum probability of bit error for given M, L, E_b/N_0, and E_b/N_J:

$$P_s(e) = \max_\gamma P_s(e;\gamma). \tag{3-18}$$

This numerical procedure was followed because of the difficulty in obtaining an analytical solution to (3-18) by differentiating the error expressions (3-9) for the square-law receiver and (3-16) for the linear-law receiver. Since the system uses orthogonal waveforms, the bit error probability is related to the symbol error probability by

$$P_b(e) = \frac{M}{2(M-1)} P_s(e). \tag{3-19}$$

In all our numerical calculations, we used 32 quantization levels.

3.4.1 Numerical Results for Square-Law Receiver

In all the results we present, the performance is the bit error probability as a function of bit-energy-to-jamming noise density ratio; for fixed M, this represents a comparison under a bit energy constraint. We have selected 10^{-5} as a practical value of the probability of bit error under jamming-free conditions for L=1 (i.e., no noncoherent combining loss). Comparison of this clipper receiver with different values of M is based on the following. For L=1, the systems corresponding to different M achieve the same bit error rate (10^{-5}) for different values of E_b/N_0; for example, for M = 2, 4, 8, 16, and 32, the required values of E_b/N_0 are 13.35 dB, 10.61 dB, 9.09 dB, 8.08 dB, and 7.33 dB, respectively. The variations in performance of the different M-ary receivers with clippers are due to their different responses to increased L and to jamming effects. The presentation of the results is organized in accordance with the parameters E_b/N_0, M, and L in the manner shown in Table 3-1.

The performance of the square-law combining receiver with clippers depends upon the choice of the clipping threshold η. In our calculations, we work with the normalized threshold η/σ^2 to avoid having to specify an absolute noise level and an absolute threshold. We define the optimum normalized threshold, η_0, as that value of η/σ^2 which minimizes $P_b(e)$ in the absence of jamming. This optimum normalized threshold η_0 is a function of M, L, and signal-to-noise ratio (E_b/N_0). Figure 3-4 depicts the optimum normalized threshold η_0 as a function of the number of hops/symbol L with M as a parameter. As L increases the optimum clipping threshold decreases; and with fixed L, higher M gives larger optimum normalized thresholds.

TABLE 3-1

SUMMARY OF FIGURES 3-4 THROUGH 3-14

FOR PERFORMANCE OF THE SQUARE-LAW COMBINING RECEIVER WITH CLIPPERS

FIGURES	PARAMETERS	CONTENTS
3-4	M = 2, 4, 8	Optimum normalized threshold vs. L
3-5 and 3-6	M = 2, 8	Optimum fraction γ_0 vs. L (E_b/N_J) as parameter
3-7	M = 8	Optimum fraction γ_0 vs. E_b/N_J (L as parameter)
3-8 and 3-9	(10^{-5} error rate without jamming) L = 2, 4	$P_b(e)$ vs. E_b/N_J (M as parameter)
3-10 through 3-12	E_b/N_0 = 10.61 dB (10^{-5} error rate without jamming) M = 4, L = 1, 2, 4	Optimum Jamming and Wideband Jamming $P_b(e)$ vs. E_b/N_J
3-13 and 3-14	(10^{-5} error rate without jamming) M = 4, 8	$P_b(e)$ vs. E_b/N_J (L as parameter)

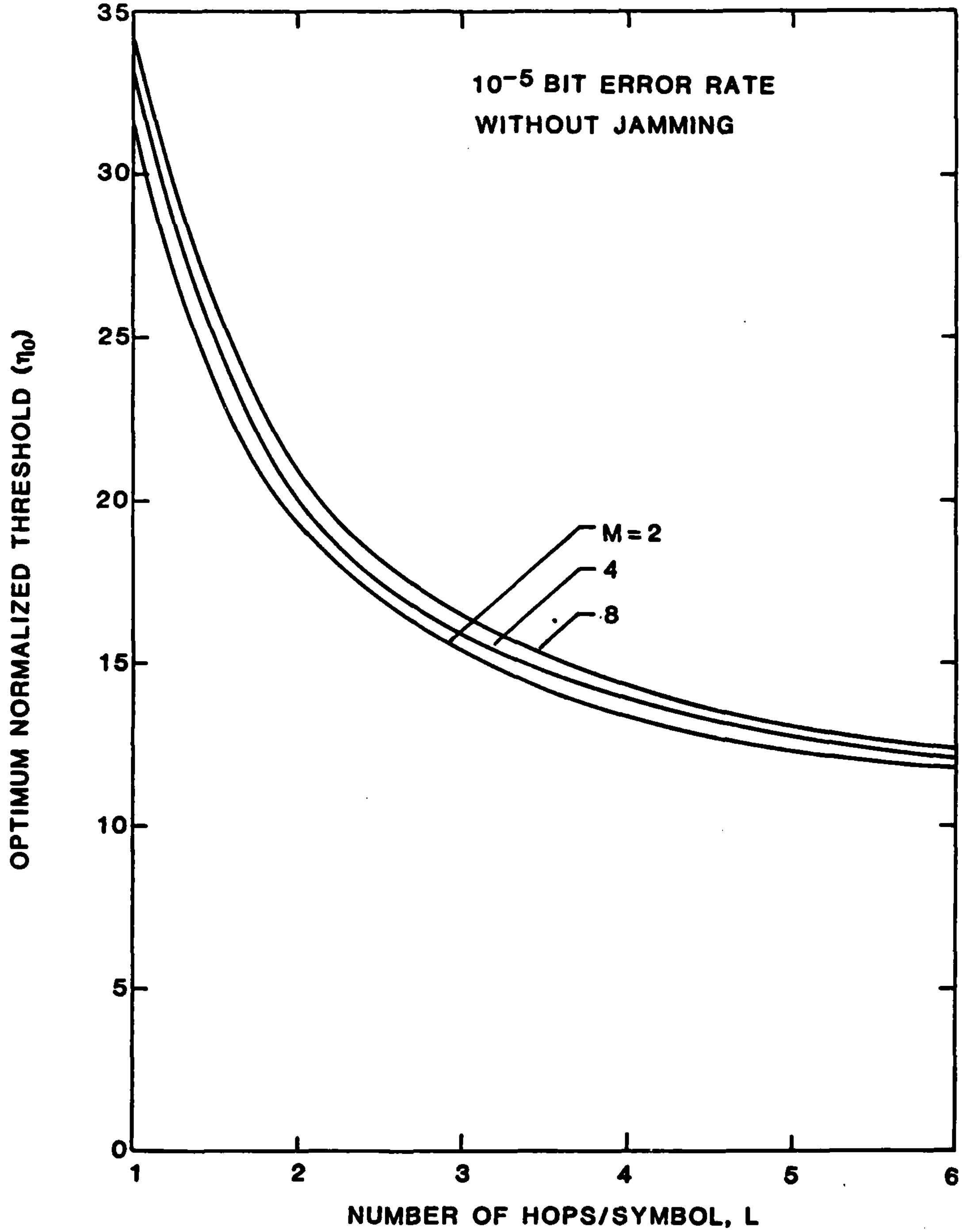


FIGURE 3-4 OPTIMUM NORMALIZED THRESHOLD η_0 VS. NUMBER OF HOPS/SYMBOL L WITH M AS A PARAMETER FOR CLIPPER RECEIVER

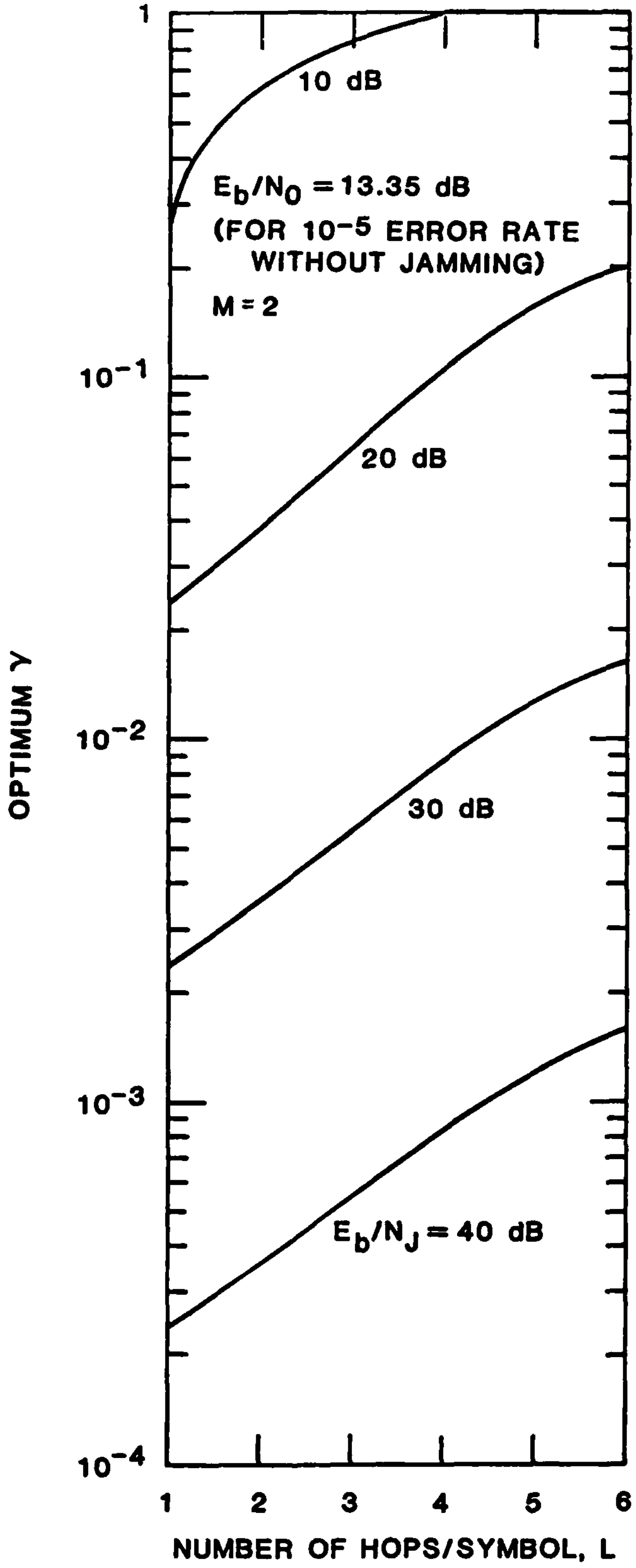


FIGURE 3-5 OPTIMUM JAMMING FRACTION (γ) VS. NUMBER OF HOPS/SYMBOL (L) FOR THE CLIPPER FH/MFSK ($M = 2$) RECEIVER WHEN $E_b/N_0 = 13.35$ dB WITH E_b/N_J AS A PARAMETER

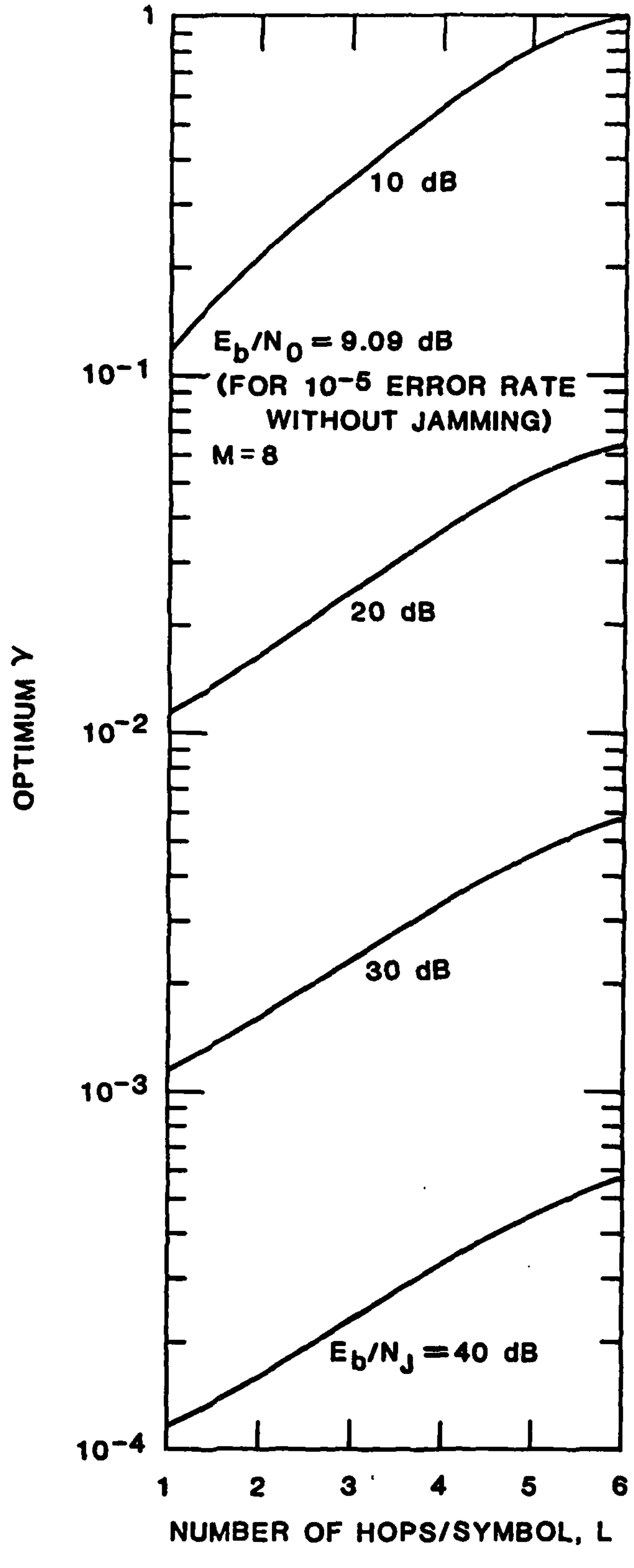


FIGURE 3-6 OPTIMUM JAMMING FRACTION (γ) VS. NUMBER OF HOPS/SYMBOL (L) FOR THE CLIPPER FH/MFSK (M = 8) RECEIVER WHEN $E_b/N_0 = 9.09$ dB WITH E_b/N_J AS A PARAMETER

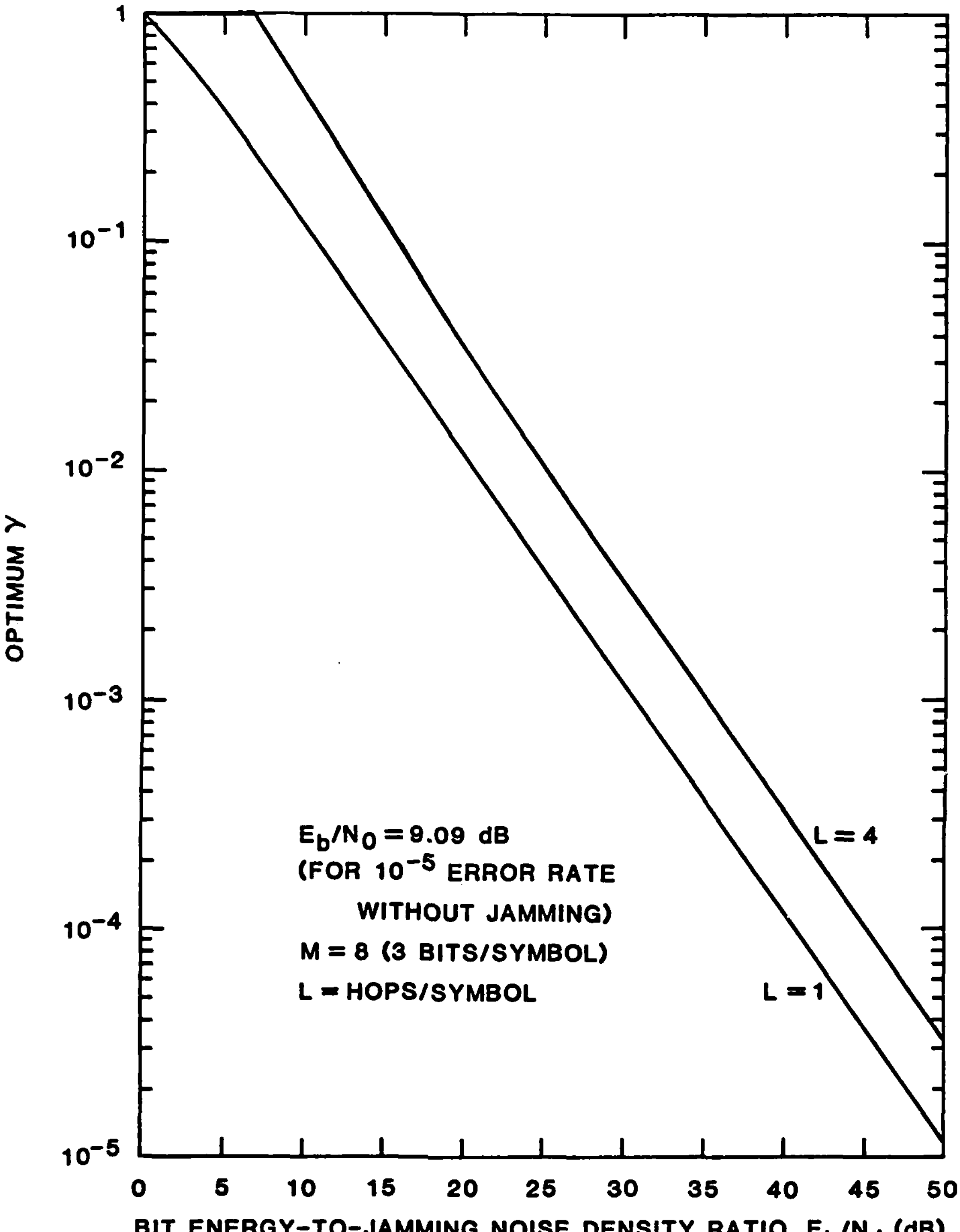


FIGURE 3-7 OPTIMUM JAMMING PERFORMANCE OF FH/MFSK (M = 8) SQUARE-LAW CLIPPER RECEIVER WHEN $E_b/N_0 = 9.09$ dB WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER

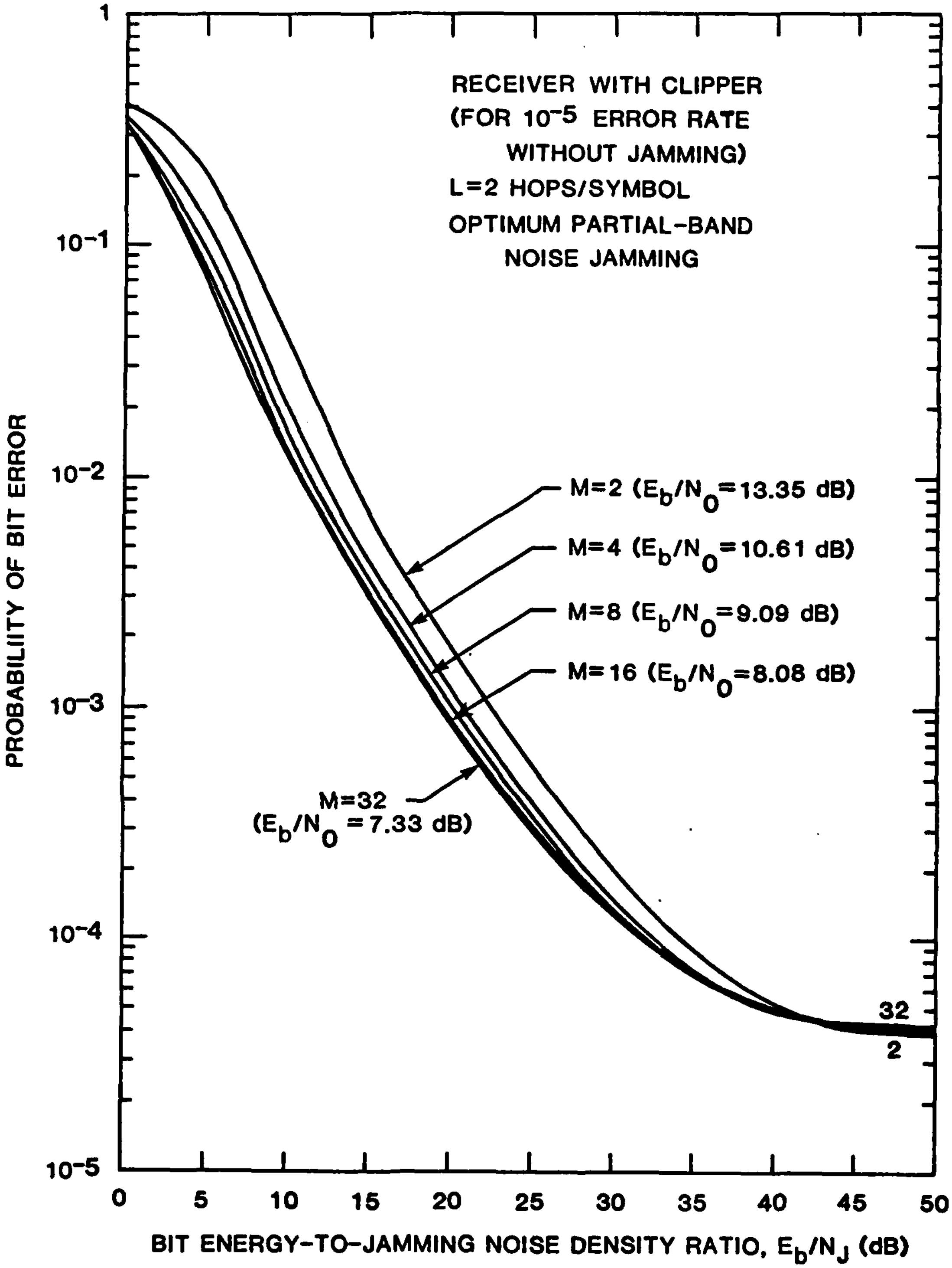


FIGURE 3-8 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/MFSK RECEIVER WITH CLIPPER FOR L=2 HOPS/SYMBOL WITH NUMBER OF SYMBOLS (M) AS A PARAMETER

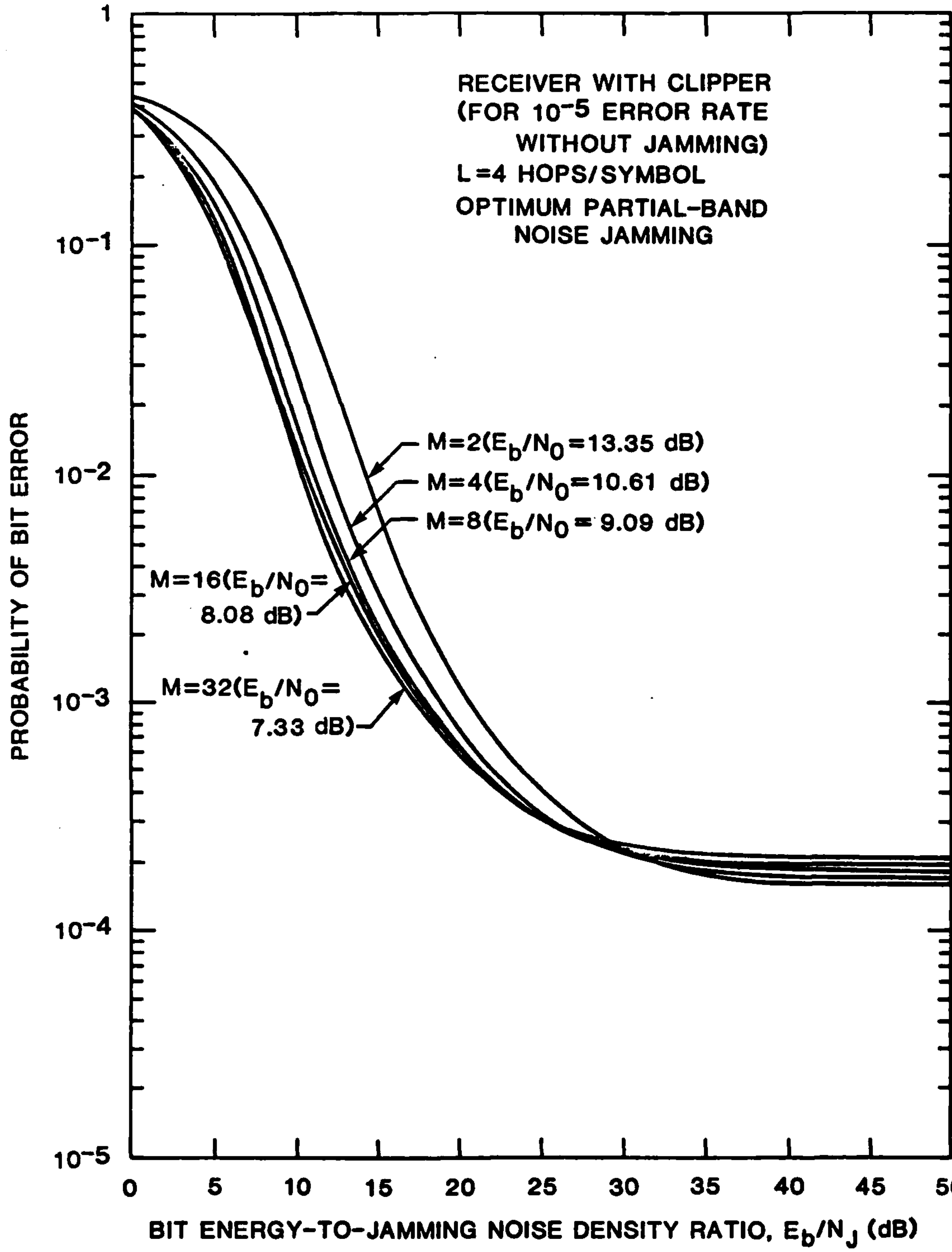


FIGURE 3-9 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/MFSK RECEIVER WITH CLIPPER FOR L=4 HOPS/SYMBOL WITH NUMBER OF SYMBOLS (M) AS A PARAMETER

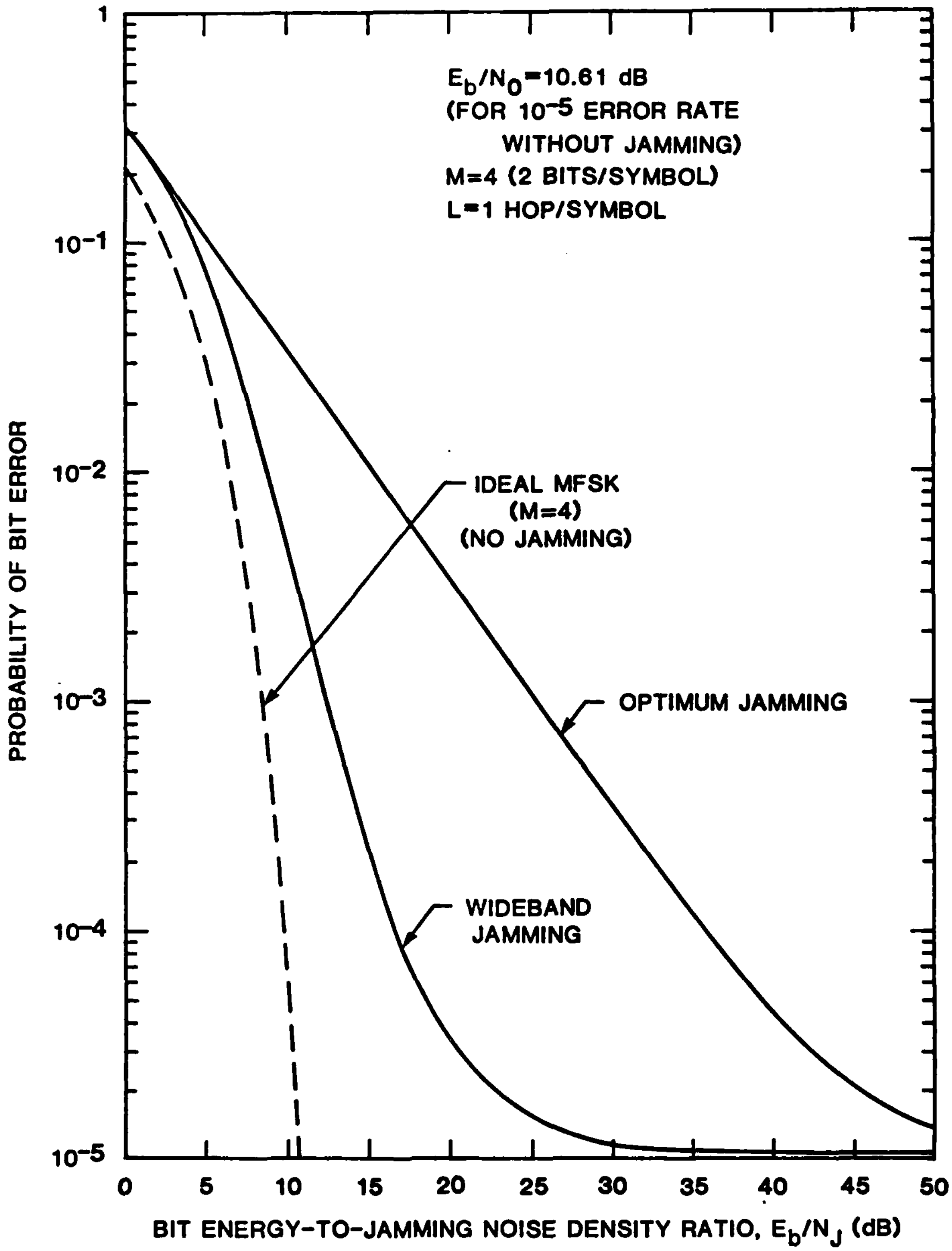


FIGURE 3-10 OPTIMUM PARTIAL-BAND NOISE JAMMING AND WIDEBAND JAMMING PERFORMANCES OF FH/MFSK (M=4) RECEIVER WITH CLIPPER FOR L=1 HOP/SYMBOL WHEN E_b/N_0=10.61 dB (FOR IDEAL MFSK (M=4) CURVE THE ABSCISSA READS E_b/N_0)

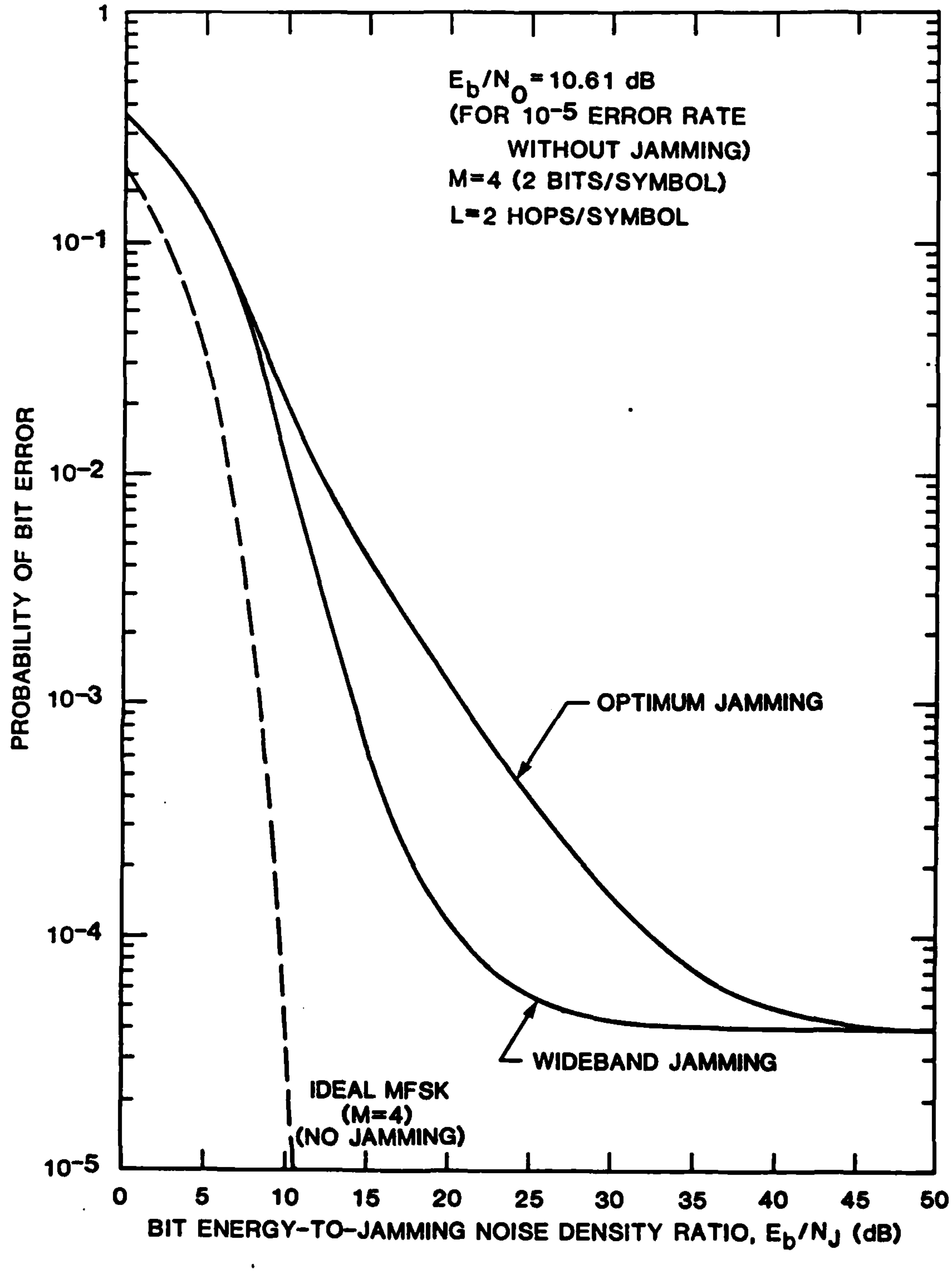


FIGURE 3-11 OPTIMUM PARTIAL-BAND NOISE JAMMING AND WIDEBAND JAMMING PERFORMANCES OF FH/MFSK (M=4) RECEIVER WITH CLIPPER FOR L=2 HOPS/SYMBOL WHEN $E_b/N_0 = 10.61$ dB (FOR IDEAL MFSK (M=4) CURVE THE ABSCISSA READS E_b/N_0)

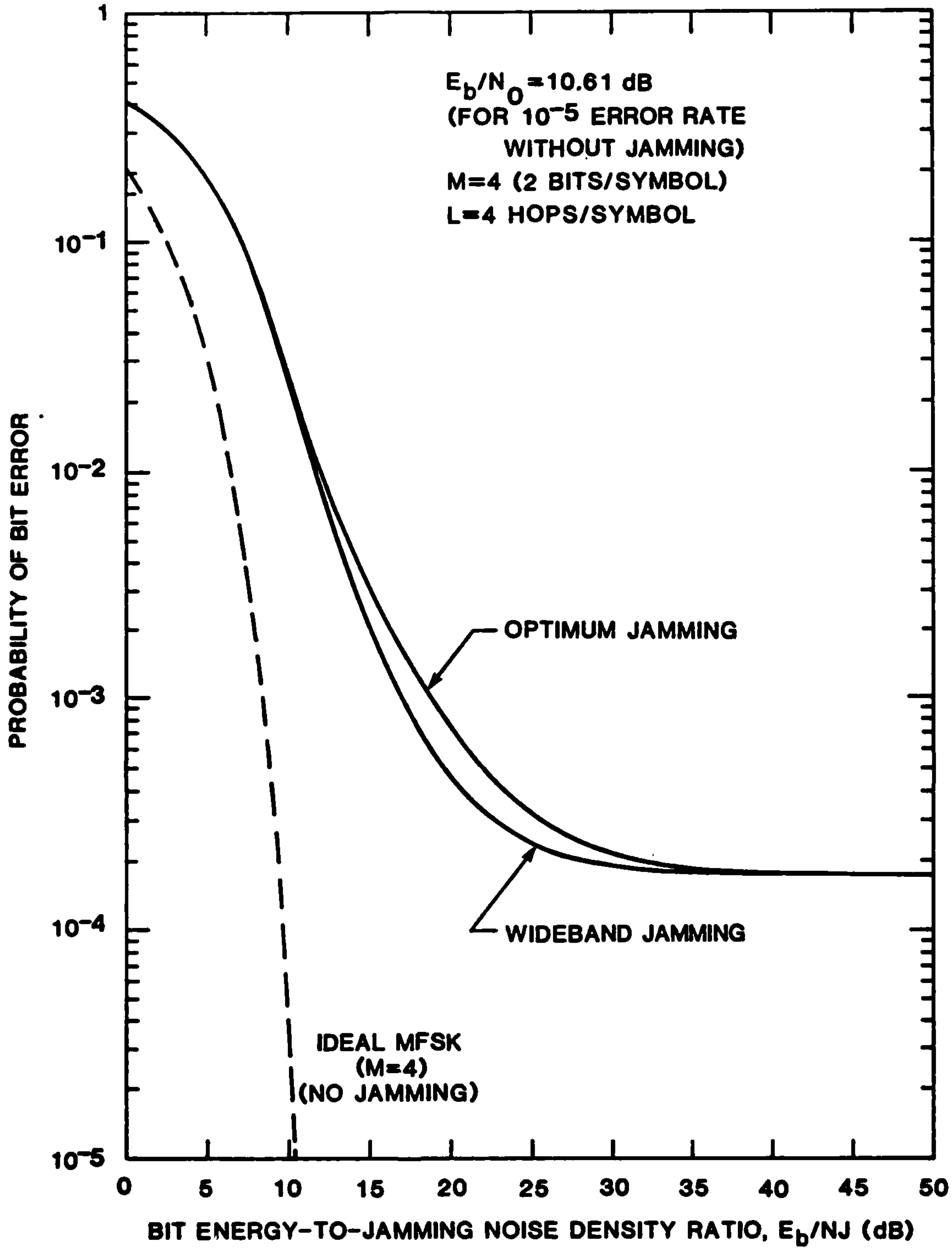


FIGURE 3-12 OPTIMUM PARTIAL-BAND NOISE JAMMING AND WIDEBAND JAMMING PERFORMANCES OF FH/MFSK (M=4) RECEIVER WITH CLIPPER FOR L=4 HOPS/SYMBOL WHEN E_b/N_O=10.61 dB (FOR IDEAL MFSK (M=4) CURVE THE ABSCISSA READS E_b/N_O)

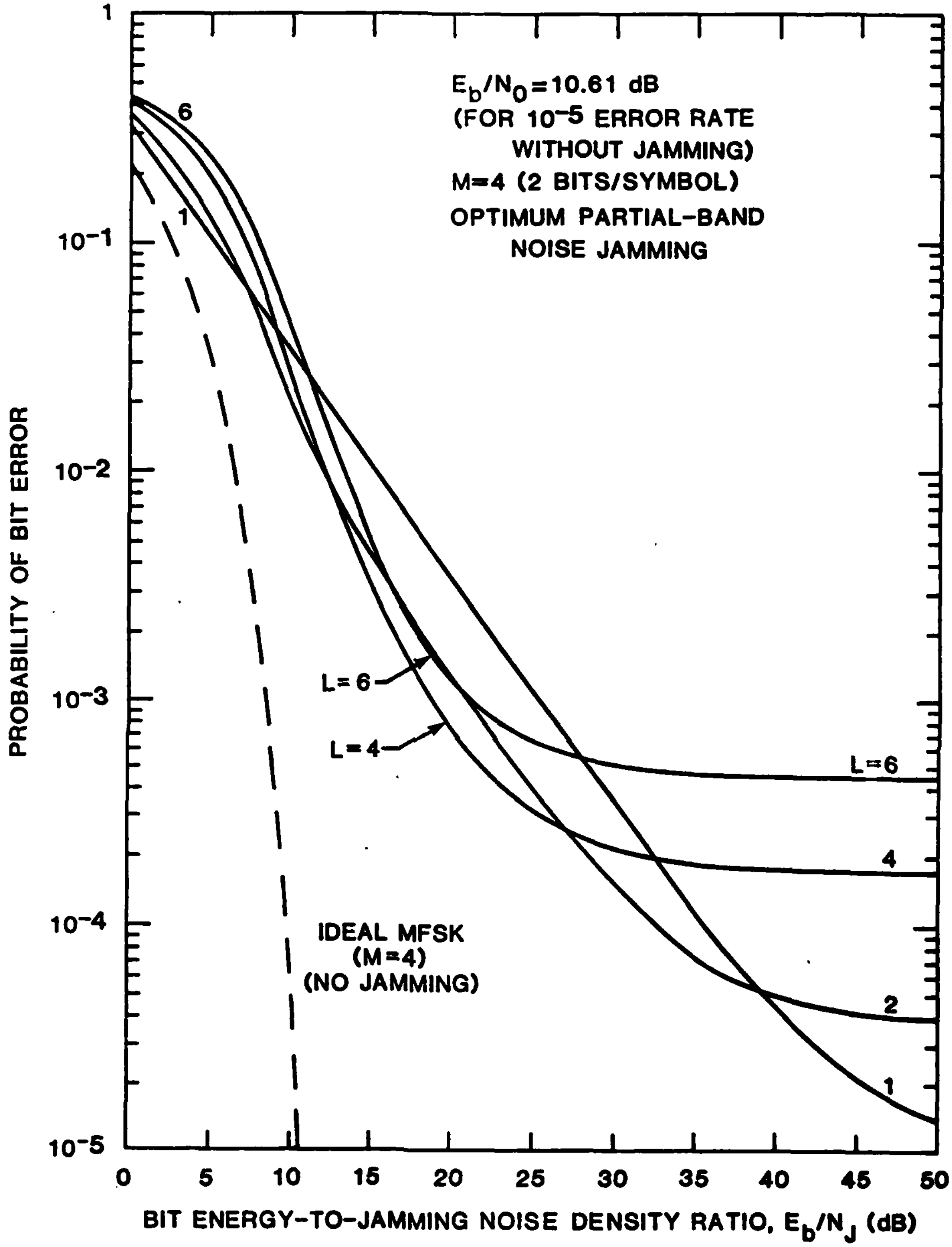


FIGURE 3-13 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/MFSK (M=4) RECEIVER WITH CLIPPER WITH NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER WHEN E_b/N_0=10.61 dB (FOR IDEAL MFSK (M=4) CURVE THE ABSCISSA READS E_b/N_0)

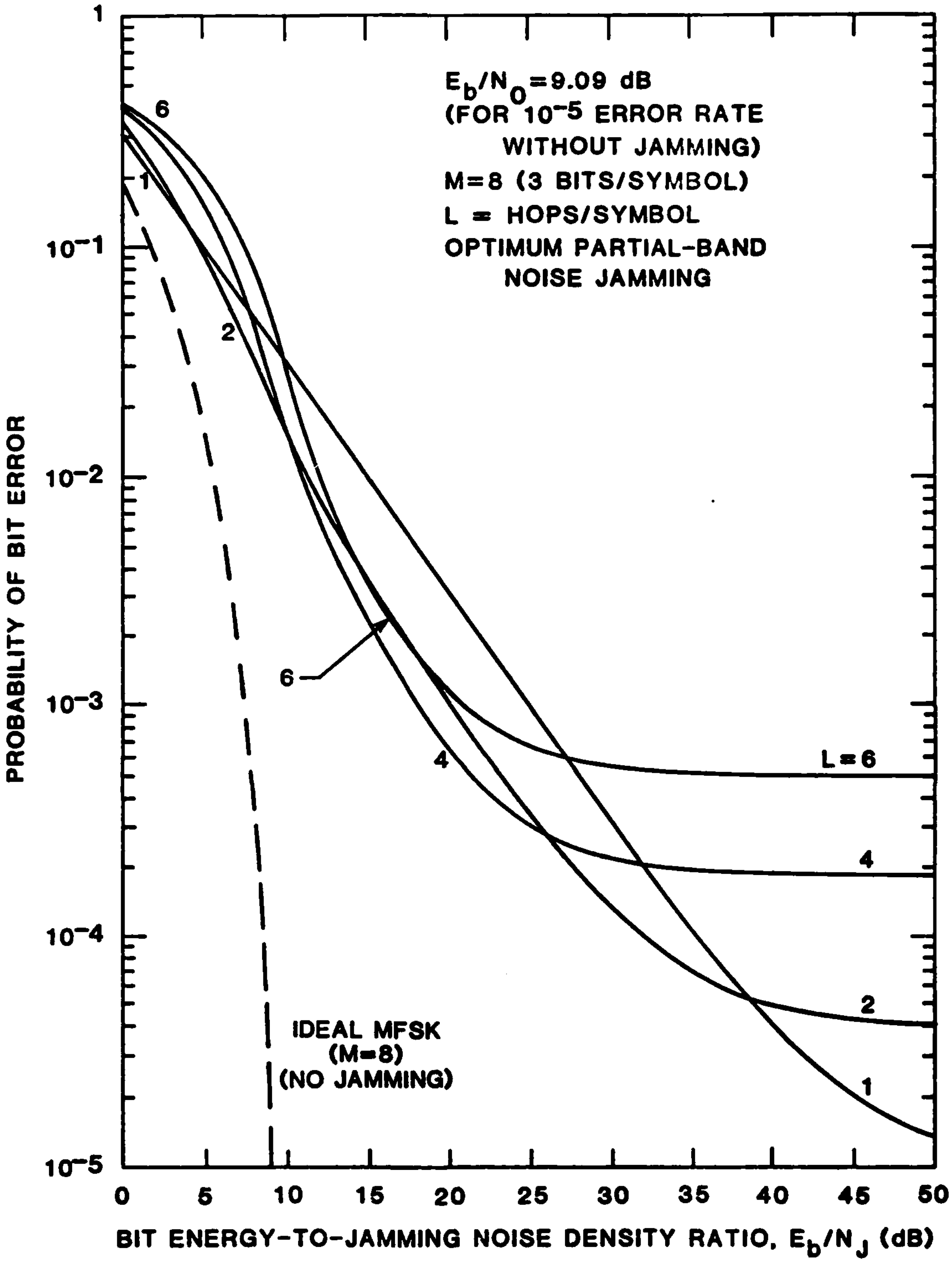


FIGURE 3-14 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/MFSK (M=8) RECEIVER WITH CLIPPER WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER WHEN E_b/N_0=9.09 dB (FOR IDEAL MFSK (M=8) CURVE THE ABSCISSA READS E_b/N_0)

Figures 3-5 and 3-6 show the typical behavior of the optimum fraction γ_0 as a function of the number of hops per symbol L for M=2 and 8. We observe that, in general, the optimum fraction γ_0 is inversely proportional to E_b/N_J when E_b/N_J exceed some value. This implies that when the available jamming power is relatively strong, then the jammer's strategy is to use wideband jamming (γ=1); and when the jamming power is weak compared to the signal, the jammer should use partial-band jamming with fraction γ_0. We observe that for this clipper receiver the value of γ_0 increases when L increases and that the slope of the curve γ_0 vs. L is nearly the same for the different values of E_b/N_J. From Figures 3-5 and 3-6 it is apparent that the jammer must acquire knowledge of M and L (especially L) in order to apply an effective optimum partial-band jamming strategy.

Figure 3-7 supports the above statements by showing the optimum fraction γ_0 vs. E_b/N_J for 8-ary FH/MFSK with different numbers of hops per symbol (L=1 and 4). The jammer's optimum strategy appears to be partial-band noise jamming with fraction γ_0, unless the available jamming power is very strong (say, E_b/N_J is less than 5 dB). The value of E_b/N_J for which γ_0 becomes less than one is a function of L.

Figures 3-8 (L=2) and 3-9 (L=4) show the probability of bit error vs. E_b/N_J for different M (M=2, 4, 8, 16, and 32). We observe that in the strong jamming region, the performance improves with increasing M and follows a nearly exponential channel behavior for these values of L. However, as E_b/N_J approaches infinity (no jamming), the performance degrades with increasing M; this behavior is due to the fact that noncoherent combining loss (NCL) effects are directly proportional to both L and M, as explained in Section 2.1.

Figures 3-10 through 3-12 compare wideband (γ=1) and optimum or worst-case partial-band noise jamming (γ_0) for M=4 and L=1, 2, and 4. The difference between wideband and optimum partial-band noise jamming is most

pronounced for L=1 hop/symbol, as illustrated in Figure 3-10. In this figure for L=1, the resulting dependency of the bit error probability is approximately inverse linear. As L increases above 1, the optimum jamming performance is improved greatly as can be observed in Figures 3-11 and 3-12. We observe that the wideband jamming performance is pushed up (or degraded) due to the NCL effect as L increases. The performance difference between the optimum jamming and wideband jamming becomes quite small for L=4 hops/symbol.

In Figures 3-13 and 3-14 we observe the tradeoff between anti-jam effectiveness and NCL which takes place as the number of hops/symbol, L, is increased. Each figure gives the bit error probability as a function of E_b/N_J for fixed M (4 and 8) and E_b/N_0 (10^{-5} error rate without jamming) with L as a parameter (L=1, 2, 4 and 6). We observe that a kind of diversity improvement is obtained for E_b/N_J between 10 dB and 40 dB. However, since L=1 always gives the best performance for high E_b/N_J, this quasi-diversity concept given by the square-law combining clipper receiver with L-hops/symbol is different from the conventional Rayleigh-fading-channel diversity concept.

3.4.2 Numerical Results for Linear-Law Receiver

In Figures 3-15 and 3-16, we compare the bit error rate performances of the FH/MFSK linear-law combining receiver with clippers and the square-law combining receiver with clippers, with L as a parameter (L=1, 2, and 4) for M=8 and 16. The figures show that the linear-law receiver provides uniformly better performance than the square-law receiver; but in the middle range of E_b/N_J (say 10 dB to 25 dB), both receivers give almost the same performance. Also, for L=1 hop/symbol, both receivers give identical results for the binary (M=2) case and almost identical results for $M > 2$.

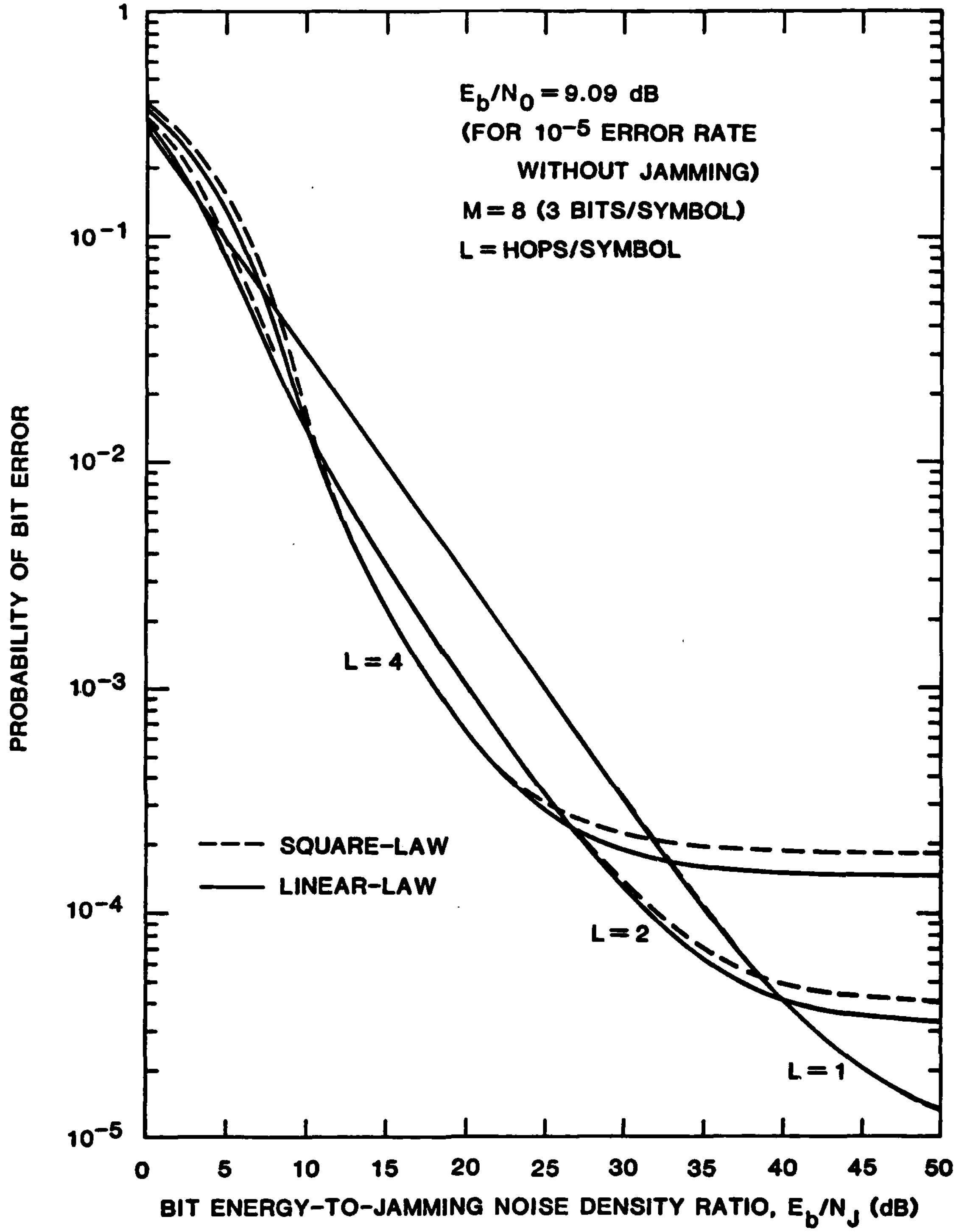


FIGURE 3-15 OPTIMUM JAMMING PERFORMANCE COMPARISONS OF FH/MFSK ($M = 8$) SQUARE-LAW COMBINING CLIPPER RECEIVER AND LINEAR-LAW COMBINING CLIPPER RECEIVER WHEN $E_b/N_0 = 9.09$ dB WITH NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER

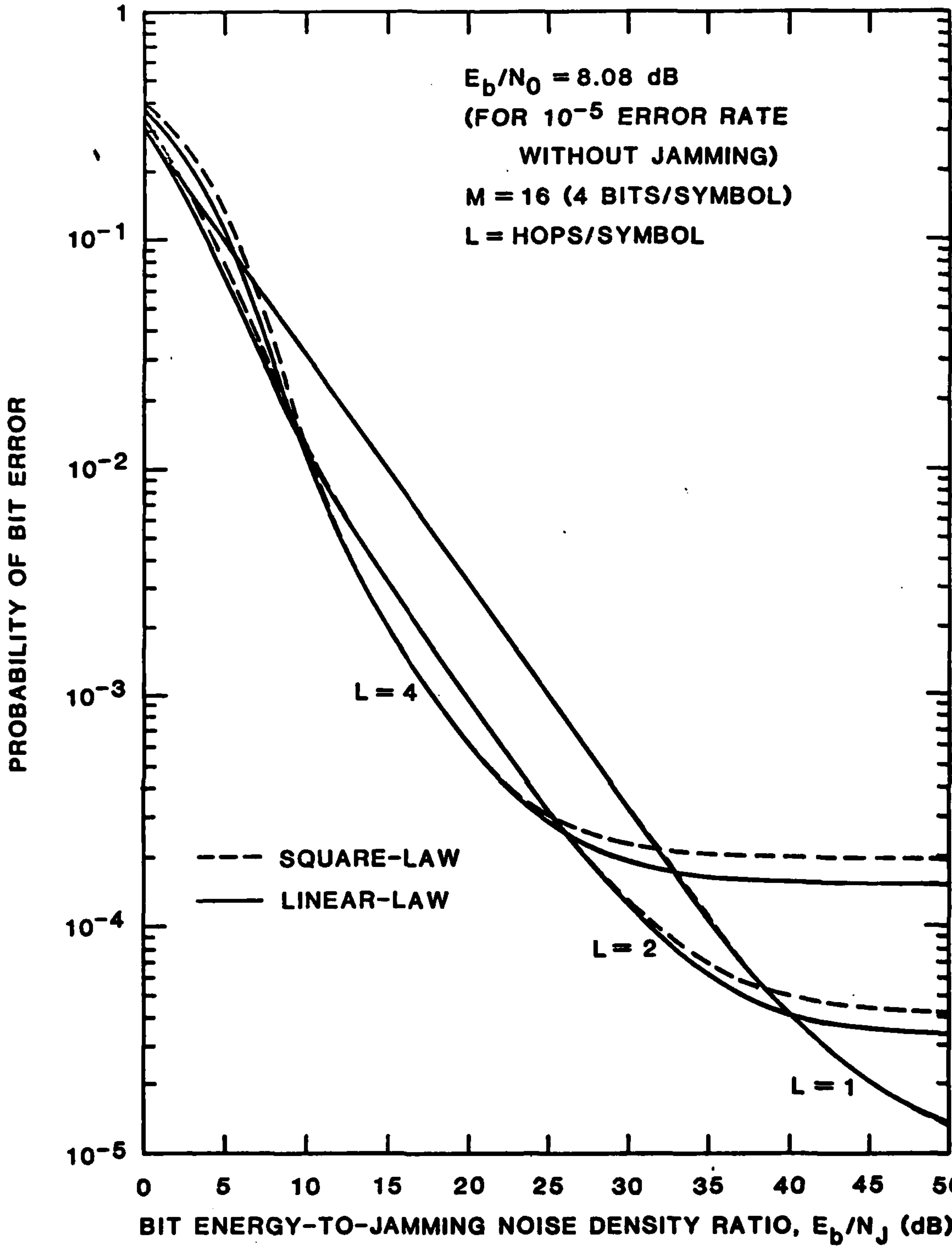


FIGURE 3-16 OPTIMUM JAMMING PERFORMANCES OF FH/MFSK (M = 16) SQUARE-LAW COMBINING CLIPPER RECEIVER AND LINEAR-LAW COMBINING CLIPPER RECEIVER WHEN E_b/N_0 = 8.08 dB WITH NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER

4.0 PARTIAL-BAND NOISE JAMMING PERFORMANCE OF AN L HOPS/SYMBOL FH/MFSK RECEIVER EMPLOYING ADAPTIVE GAIN CONTROL

In Section 2 we considered a conventional FH/MFSK receiver with L hops/symbol in which the symbol decision is based on linear combining of the square-law detected hops. It was shown that the bit error probability for the system always increases when L increases; there is no diversity gain associated with using L hops/symbol.

When the standard FH/MFSK receiver is modified by inserting clippers (soft limiters) prior to accumulating the envelope detector outputs, as shown in Section 3, the system performance against optimum partial-band noise jamming improves greatly. A kind of limited diversity gain is exhibited in which increasing L reduces the error probability for certain values of E_b/N_J and E_b/N_0, the ratios of bit energy to jammer noise density and to thermal noise density, respectively.

We now consider another modification to the standard FH/MFSK receiver in which the detector outputs are normalized by the received noise power on a per-hop basis, as illustrated in Figure 4-1. The message symbol decision $\hat{m}$ is taken to be the index of the largest of the decision statistics z_i, where

$$z_i = \sum_{k=1}^{L} z_{ik} = \sum_{k=1}^{L} x_{ik}/\sigma_k^2, \quad i = 1, 2, \ldots, M; \tag{4-1}$$

x_{ik} is the sampled squared envelope in channel i on the kth hop; and σ_k^2 is the variance or average received noise power on the kth hop. The design is idealized in that it is predicated on the assumption that the noise power on a given hop is measured perfectly (using a separate channel as shown in Figure 4-1) and is the same for all channels. Because of this ideal

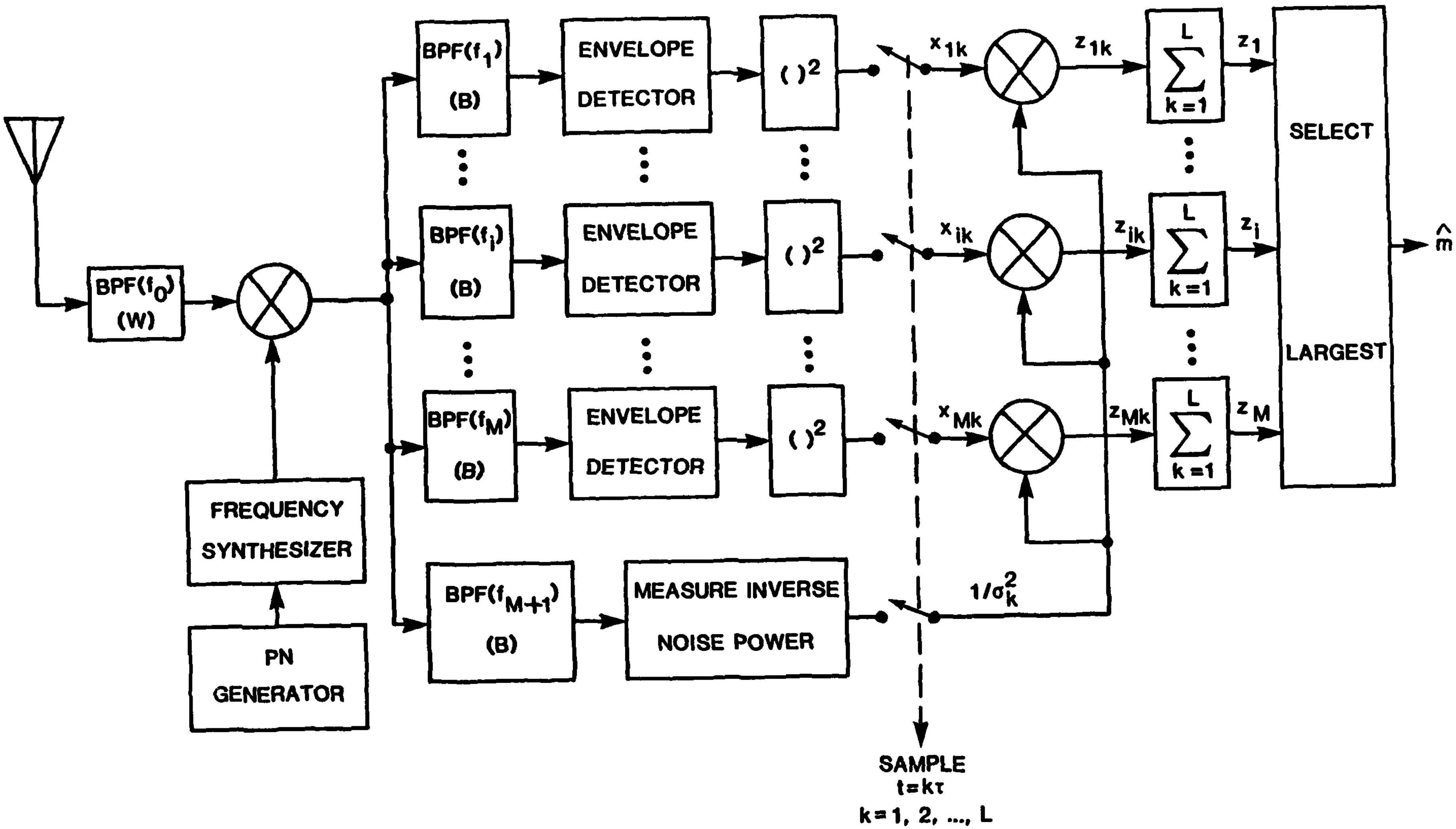


FIGURE 4-1 FH/MFSK RECEIVER WITH ADAPTIVE GAIN CONTROL FOR USE IN PARTIAL-BAND JAMMING ENVIRONMENT

adaptive gain control (AGC) normalization, analysis of the performance of the receiver is expected to be useful as a lower bound on what may be realized in practice. In Section 4.4 we consider also the receiver of Figure 4-1 when linear-law envelope detection is used.

4.1 SIGNAL, NOISE, AND JAMMING MODELS

After dehopping, the received signal is assumed equally likely to be present in any one of the M channels for the entire symbol period $T_s = L\tau$, where τ is the hop period and L is the number of hops per MFSK symbol. Without loss of generality, we assume that the signal with power S is in channel 1, or

$$s(t) = \sqrt{2S}\cos(\omega_1 t + \theta_k),\ (k-1)\tau < t \leq k\tau,\quad k = 1, 2, \ldots, L. \tag{4-2}$$

Thermal noise is considered also to be present in each channel, and is assumed to be zero-mean narrowband Gaussian noise with variance $\sigma_N^2 = N_0 B$, where $N_0/2$ is the (two-sided) noise power spectral density and B is the bandwidth of each channel. Thus for no jamming the samples of the M squared envelope detector outputs on the kth hop are the variables

$$x_{1k} = \left(\sqrt{2S}\cos\theta_k + n_{c1k}\right)^2 + \left(\sqrt{2S}\sin\theta_k + n_{s1k}\right)^2 \tag{4-3a}$$

and

$$x_{ik} = n_{cik}^2 + n_{sik}^2,\quad i = 2, 3, \ldots, M, \tag{4-3b}$$

where n_{cik}, n_{sik}, $i = 1, 2, \ldots, M$; $k = 1, 2, \ldots, L$, are the independent noise quadrature components in the channels at the sample times $t_k = k\tau$, with

$$E\left\{n_{cik}^2\right\} = E\left\{n_{sik}^2\right\} = \sigma_N^2 = N_0 B,\quad \text{for all } i, k. \tag{4-4}$$

Jamming noise is assumed to be present on a given hop with probability γ. When jamming occurs, it is considered to be in all of the M channels (as well as the measurement channel). This model assumes that if the total number of hopped frequency "slots" is N_1, making the system bandwidth $W = N_1 B$, the jammer's bandwidth is $N_2 B$ where $N_2 \leq N_1$. Under these conditions, the number of possible hop positions for the MFSK bandwidth (including the measurement channel) is $N_1 - M$, and the jamming probabilities associated with these positions are

$$\pi_1 = \Pr\{\text{all } M+1 \text{ slots jammed}\} = \frac{N_2 - M}{N_1 - M}, \tag{4-5a}$$

$$\pi_0 = \Pr\{\text{none of the } M+1 \text{ slots jammed}\} = \frac{N_1 - N_2 - 2M}{N_1 - M}, \tag{4-5b}$$

and

$$\pi_p = \Pr\{\text{some of the } M+1 \text{ slots jammed}\} = \frac{2M}{N_1 - M}. \tag{4-5c}$$

Now, defining $\gamma \triangleq N_2/N_1$ and $\beta \triangleq M/N_1$, we see that

$$\left.\begin{aligned} \pi_1 &= \frac{\gamma-\beta}{1-\beta} \approx \gamma \\ \pi_0 &= \frac{1-\gamma-2\beta}{1-\beta} \approx 1-\gamma \\ \pi_p &= \frac{2\beta}{1-\beta} \approx 0 \end{aligned}\right\} \quad \beta \ll 1 \text{ or } M \ll N_1 . \tag{4-6}$$

Thus for very wide system bandwidth, we may ignore the possibility that only some of the MFSK slots are jammed, and take

$$\gamma = \Pr\{\text{symbol jammed}\} = \frac{\text{Jammer bandwidth}}{\text{System bandwidth}} . \tag{4-7}$$

When jamming noise is present, in each channel it is assumed to be zero-mean narrowband Gaussian noise with variance $\sigma_J^2 = N_J B/\gamma$, where $N_J/2$ is the (two-sided) noise power spectral density averaged over the system

bandwidth; that is,

$$N_J = \frac{J}{W}. \tag{4-8}$$

The combination of jamming and thermal noise on the kth hop produces the detector output samples

$$x_{1k} = \left(\sqrt{2S}\cos\theta_k + n_{c1k} + j_{c1k}\right)^2 + \left(\sqrt{2S}\sin\theta_k + n_{s1k} + j_{s1k}\right)^2 \tag{4-9a}$$

$$x_{ik} = \left(n_{cik} + j_{cik}\right)^2 + \left(n_{sik} + j_{sik}\right)^2, \quad i = 2, 3, \ldots, M, \tag{4-9b}$$

where j_{cik}, j_{sik}, $i = 1, 2, \ldots, M$; $k = 1, 2, \ldots, L$, are the independent jamming noise quadrature components in the channels at the sample times, with

$$E\left\{j_{cik}^2\right\} = E\left\{j_{sik}^2\right\} = \sigma_J^2 = N_J B/\gamma, \quad \text{for all } i, k. \tag{4-10}$$

In summary, we can express the detector output samples as

$$x_{1k} = \sigma_k^2\left[\left(\sqrt{\frac{2S}{\sigma_k^2}}\cos\theta_k + \nu_{c1k}\right)^2 + \left(\sqrt{\frac{2S}{\sigma_k^2}}\sin\theta_k + \nu_{s1k}\right)^2\right] \tag{4-11a}$$

$$x_{ik} = \sigma_k^2\left(\nu_{cik}^2 + \nu_{sik}^2\right), \quad i = 2, 3, \ldots, M, \tag{4-11b}$$

where ν_{cik} and ν_{sik} are independent unit-variance zero-mean Gaussian random variables and

$$\sigma_k^2 = \begin{cases} \sigma_N^2 = N_0 B & \text{with probability } 1-\gamma \\ \sigma_T^2 = \sigma_N^2 + \sigma_J^2 = (N_0 + N_J/\gamma)B & \text{with probability } \gamma. \end{cases} \tag{4-11c}$$

4.2 PROBABILITY OF ERROR ANALYSIS FOR SQUARE-LAW AGC RECEIVER

Assuming equally likely M-ary symbols, we may express the symbol error probability by

$$P_s(e) = P_s(e|m_1 \text{ transmitted})$$

$$= \sum_{\ell=0}^{L} p_\ell \cdot P_s(e|m_1, \ell \text{ hops jammed}), \tag{4-12}$$

where p_ℓ is the probability that ℓ out of L hops are jammed:

$$p_\ell = \binom{L}{\ell} \gamma^\ell (1-\gamma)^{L-\ell}. \tag{4-13}$$

The summation in (4-12) represents averaging the conditional symbol error probability over the possible jamming events.

For M a power of two ($M=2^K$), the bit error probability is obtained from the symbol error probability using the relation

$$P_b(e) = \frac{M/2}{M-1} P_s(e). \tag{4-14}$$

4.2.1 Distribution of the Decision Statistics

From (4-1), the decision statistics z_i are defined as

$$z_1 = \sum_{k=1}^{L} x_{1k}/\sigma_k^2$$

$$= \sum_{k=1}^{L}\left[\left(\sqrt{\frac{2S}{\sigma_k^2}}\cos\theta_k + \nu_{c1k}\right)^2 + \left(\sqrt{\frac{2S}{\sigma_k^2}}\sin\theta_k + \nu_{s1k}\right)^2\right] \tag{4-15a}$$

and

$$z_i = \sum_{k=1}^{L} x_{ik}/\sigma_k^2$$

$$= \sum_{k=1}^{L} \left(\nu_{cik}^2 + \nu_{sik}^2\right), \quad i = 2, 3, \ldots, M. \tag{4-15b}$$

Since ν_{cik} and ν_{sik} are independent zero-mean unit-variance Gaussian random variables, conditionally the z_i for $i \geq 2$ are chi-squared random variables with 2L degrees of freedom and z_1 is a noncentral chi-squared random variable with 2L degrees of freedom and noncentrality parameter

$$\lambda_\ell \equiv 2\rho_\ell \triangleq \sum_{k=1}^{L}\left[\left(\sqrt{\frac{2S}{\sigma_k^2}}\cos\theta_k\right)^2 + \left(\sqrt{\frac{2S}{\sigma_k^2}}\sin\theta_k\right)^2\right]$$

$$= 2\sum_{k=1}^{L} \frac{S}{\sigma_k^2} = \ell\,\frac{2S}{\sigma_T^2} + (L-\ell)\,\frac{2S}{\sigma_N^2}\,. \tag{4-16}$$

Thus the probability density function for z_1 is

$$p_{z_1}(\beta;\lambda_\ell) = \frac{1}{2}\left(\frac{\beta}{\lambda_\ell}\right)^{(L-1)/2} e^{-(\beta+\lambda_\ell)/2}\; I_{L-1}\left(\sqrt{\beta\lambda_\ell}\right)$$

$$= e^{-\rho_\ell} \sum_{m=0}^{\infty} \frac{\rho_\ell^m}{m!}\cdot\frac{1}{2}\, e^{-\beta/2}\, \frac{(\beta/2)^{m+L-1}}{\Gamma(m+L)}\,, \quad \beta > 0 \tag{4-17}$$

where $I_{L-1}(\cdot)$ is the modified Bessel function of the first kind and order L-1. The pdf's for the noise-only channel statistics z_i $(i \geq 2)$ are identical and are given by

$$p_{z_i}(\alpha) = p_{z_2}(\alpha) = \frac{1}{2}\, e^{-\alpha/2}\, \frac{(\alpha/2)^{L-1}}{\Gamma(L)}\,, \quad \alpha > 0,\ i = 2, 3, \ldots, M. \tag{4-18}$$

4.2.2 Conditional Symbol Error Probability

Since for $M > 2$ there are many error events but only one correct decision, it is convenient to write the conditional symbol error probability as

$$\begin{aligned} P_s(e|\ell) &\equiv P_s(e|\ell \text{ hops jammed}) \\ &= 1 - P_s(c|m_1, \ell) \\ &= 1 - \Pr\left\{z_2 < z_1, z_3 < z_1, \ldots, z_M < z_1\right\}. \end{aligned} \quad (4\text{-}19)$$

In terms of the pdf's for the statistics, this becomes

$$P_s(e|\ell) = 1 - \int_0^\infty d\beta \; p_{z_1}(\beta;\lambda_\ell)\left[\int_0^\beta d\alpha \; p_{z_2}(\alpha)\right]^{M-1} . \quad (4\text{-}20)$$

From (4-18) we find that

$$\begin{aligned} \int_0^\beta d\alpha \; p_{z_2}(\alpha) &= 1 - \int_\beta^\infty d\alpha \cdot \frac{1}{2} \; e^{-\alpha/2} \; \frac{(\alpha/2)^{L-1}}{\Gamma(L)} \\ &= 1 - e^{-\beta/2} \sum_{r=0}^{L-1} \frac{(\beta/2)^r}{r!} \\ &= 1 - e^{-\beta/2} \; e_{L-1}(\beta/2), \end{aligned} \quad (4\text{-}21)$$

where $e_{L-1}(\cdot)$ is the incomplete exponential function [4, eq. 6.5.11]. The error expression (4-20) requires that (4-21) be raised to the M-1 power. Using the binomial expansion, this power is

$$\left[1 - e^{-\beta/2}\, e_{L-1}(\beta/2)\right]^{M-1} = \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m\, e^{-m\beta/2} \left[e_{L-1}(\beta/2)\right]^m . \tag{4-22}$$

Substituting (4-22) into (4-20) yields

$$P_s(e|\ell) = \sum_{m=1}^{M-1} \binom{M-1}{m} (-1)^{m+1} \int_0^\infty d\beta\, p_{z_1}\left(\beta;\lambda_\ell\right) e^{-m\beta/2} \left[e_{L-1}(\beta/2)\right]^m, \tag{4-23a}$$

in which the integral is

$$\int_0^\infty \frac{d\beta}{2}\cdot\left(\frac{\beta}{\lambda_\ell}\right)^{(L-1)/2} e^{-(\beta+\lambda_\ell)/2}\, I_{L-1}\left(\sqrt{\beta\lambda_\ell}\right) e^{-m\beta/2} \left[e_{L-1}(\beta/2)\right]^m$$

$$= e^{-\rho_\ell} \left(\frac{1}{\sqrt{\rho_\ell}}\right)^{L-1} \int_0^\infty dx\, e^{-(m+1)x}\, x^{(L-1)/2}\, I_{L-1}\left(2\sqrt{x\rho_\ell}\right) \left[e_{L-1}(x)\right]^m. \tag{4-23b}$$

Since $e_{L-1}(x)$ is an $(L-1)$-degree polynomial in x, raising it to the mth power produces an $m(L-1)$-degree polynomial:

$$\left[e_{L-1}(x)\right]^m = \sum_{r=0}^{m(L-1)} c_r(m,L)\, x^r/r!, \tag{4-24a}$$

where, from Appendix 4A, the coefficients c_r are given by

$$c_r(m,L) = \begin{cases} m^r, \quad 0 \le r \le L-1 & \text{(4-24b)} \\ \dfrac{1}{r} \displaystyle\sum_{n=1}^{L-1} \binom{r}{n} [(m+1)n-r]\, c_{r-n}(m,L), \quad r > L-1. & \text{(4-24c)} \end{cases}$$

Using (4-24a) in (4-23b) gives

$$e^{-\rho_\ell} \left(\frac{1}{\sqrt{\rho_\ell}}\right)^{L-1} \sum_{r=0}^{m(L-1)} \frac{c_r(m,L)}{r!} \int_0^\infty dx\, e^{-(m+1)x}\, x^{r+(L-1)/2}\, I_{L-1}\left(2\sqrt{x\rho_\ell}\right)$$

$$= \frac{1}{(m+1)^L} \exp\left\{-\frac{m}{m+1}\rho_\ell\right\} \sum_{r=0}^{m(L-1)} \frac{c_r(m,L)}{(m+1)^r}\, \mathcal{L}_r^{(L-1)}\left(\frac{-\rho_\ell}{m+1}\right) \tag{4-25}$$

where $\pounds_n^m(x)$ is the generalized Laguerre polynomial. In writing (4-25) we have used formulas 6.643.2, 9.220.2, and 8.972.1 in [2].

4.2.3 Bit Error Probability

Making the appropriate substitutions in (4-14) and (4-12) yields the bit error probability expression for the AGC FH/MFSK receiver:

$$P_b(e) = \frac{1}{2}\cdot\frac{M}{M-1} \sum_{\ell=0}^{L}\binom{L}{\ell}\gamma^{\ell}(1-\gamma)^{L-\ell}\sum_{m=1}^{M-1}\binom{M-1}{m}\frac{(-1)^{m+1}}{(m+1)^L}\exp\left\{\frac{-m}{m+1}\rho_\ell\right\}$$

$$\times \sum_{r=0}^{m(L-1)} \frac{c_r(m,L)}{(m+1)^r}\,\pounds_r^{(L-1)}\left(\frac{-\rho_\ell}{m+1}\right), \tag{4-26a}$$

$$= \frac{1}{2}\cdot\frac{M}{M-1} \sum_{\ell=0}^{L}\binom{L}{\ell}\gamma^{\ell}(1-\gamma)^{L-\ell}\sum_{m=1}^{M-1}\binom{M-1}{m}\frac{(-1)^{m+1}}{m+1}\exp\left\{-\frac{m}{m+1}\rho_\ell\right\} h(\rho_\ell;\, m,L), \tag{4-26b}$$

where $c_r(m,L)$ is given by (4-24),

$$\rho_\ell = \ell\cdot\frac{S}{\sigma_T^2} + (L-\ell)\frac{S}{\sigma_N^2}, \tag{4-27}$$

and we define the $m(L-1)$-degree polynomials $h(x;\, m,\, L)$ as

$$h(x;\, m,\, L) \triangleq \left(\frac{1}{m+1}\right)^{L-1}\sum_{r=0}^{m(L-1)}\frac{c_r(m,L)}{(m+1)^r}\,\pounds_r^{(L-1)}\left(-\frac{x}{m+1}\right) \tag{4-28a}$$

$$= 1 + \beta_2 x + \beta_3 x^2 + \ldots + \beta_{m(L-1)}x^{m(L-1)}. \tag{4-28b}$$

These polynomials are generalizations of the $g(x;L)$ polynomials used in [1] for the case M=2; in fact, $h(x;1,L) = g(x/2;\, L)$. For L=1, $h(x;\, m,\, 1) \equiv 1$; therefore, all the effects of $L > 1$ are embedded in these polynomial factors.

So far the expressions we have developed involve power signal-to-noise ratios. Conversion to energy and noise power spectral densities is possible using the relations

$$S = \frac{E_s}{L\tau} = \frac{K}{L} \cdot \frac{E_b}{\tau}\ , \ (K = \log_2 M); \tag{4-29a}$$

$$\sigma_N^2 = N_0 B,\ \sigma_J^2 = N_J B/\gamma; \tag{4-29b}$$

and

$$B\tau = 1. \tag{4-29c}$$

Using these relations, we have

$$\rho_\ell = \frac{E_s}{N_0} \cdot \frac{1}{L}\left[\ell \cdot \frac{\gamma a}{1+\gamma a} + (L-\ell)\right] \tag{4-30a}$$

$$= \frac{E_b}{N_0} \cdot \frac{K}{L}\left[L - \frac{\ell}{1+\gamma a}\right] \tag{4-30b}$$

$$= \frac{K}{L}\left[\ell\, \frac{E_b}{N_T} + (L-\ell)\, \frac{E_b}{N_0}\right] \tag{4-30c}$$

in which we define the parameters

$$a \triangleq N_0/N_J = \left(\frac{E_b}{N_J}\right) \Big/ \left(\frac{E_b}{N_0}\right) \tag{4-30d}$$

and

$$N_T \triangleq N_0 + N_J/\gamma\ . \tag{4-30e}$$

In terms of these new quantities, the bit error probability (4-26) becomes, for $M = 2^K$,

$$P_b(e) = P_b\left(e;\ M,\ K,\ \gamma,\ \frac{E_b}{N_0}\ ,\ \frac{E_b}{N_J}\right)$$

$$= \frac{1}{2} \cdot \frac{M}{M-1} \sum_{\ell=0}^{L} \binom{L}{\ell} \gamma^\ell (1-\gamma)^{L-\ell} \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m+1}}{m+1}$$

$$\cdot \exp\left\{-\frac{m}{m+1}\frac{K}{L}\cdot\left[\ell\frac{E_b}{N_T}+(L-\ell)\frac{E_b}{N_0}\right]\right\}\cdot h\left\{\frac{K}{L}\left[\ell\frac{E_b}{N_T}+(L-\ell)\frac{E_b}{N_0}\right];\ m,\ L\right\}. \tag{4-31}$$

In the numerical computations which follow in Section 4.3, we are interested in the error rate given by (4-31) as maximized with respect to γ, the partial-band jamming fraction.

4.2.4 Special Cases

The bit error probability expression can be better understood and checked by considering some special cases.

4.2.4.1 One Hop Per Symbol (L=1)

For one hop per symbol (4-31) reduces to

$$P_b(e;L=1) = (1-\gamma)\cdot P_M\left(\frac{E_b}{N_0}\right) + \gamma\, P_M\left(\frac{E_b}{N_0+N_J/\gamma}\right) \tag{4-32a}$$

where $P_M(\cdot)$ is the usual M-ary orthogonal error expression [7, p. 577],

$$P_M(x) = \frac{1}{2}\cdot\frac{M}{M-1}\sum_{m=1}^{M-1}\binom{M-1}{m}\frac{(-1)^{m+1}}{m+1}\exp\left\{-\frac{m}{m+1}Kx\right\}. \tag{4-32b}$$

The error rate in this case is simply the average of two error rates, one for the unjammed SNR, and the other for the jammed SNR.

4.2.4.2 Binary FSK (M=2)

For BFSK (M=2), (4-31) reduces to

$$P_b(e;M=2) = \sum_{\ell=0}^{L}\binom{L}{\ell}\gamma^{\ell}(1-\gamma)^{L-\ell}\cdot\frac{1}{2}\exp\left\{-\frac{1}{2}\cdot\frac{1}{L}\left[\ell\frac{E_b}{N_T}+(L-\ell)\frac{E_b}{N_0}\right]\right\}$$

$$\cdot\, g\left\{\frac{1}{2}\cdot\frac{1}{L}\left[\ell\frac{E_b}{N_T}+(L-\ell)\frac{E_b}{N_0}\right];\ L\right\}, \tag{4-33}$$

which agrees with the results shown in [1, p. 225], [8].

4.2.4.3 Two Hops Per Symbol (L=2)

For two hops per symbol the error probability expression is not too unwieldy towrite out, giving

$$P_b(e;L=2) = \frac{1}{2}\cdot\frac{M}{M-1}\ (1-\gamma)^2 \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m+1}}{m+1} \exp\left[-\left(\frac{m}{m+1}\right)\cdot\left(\frac{KE_b}{N_0}\right)\right]$$

$$\cdot\ h\left(\frac{KE_b}{N_0}\ ;\ m,2\right)$$

$$+\ 2\gamma(1-\gamma) \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m+1}}{m+1} \exp\left[-\frac{m}{m+1}\cdot\frac{K}{2}\left(\frac{E_b}{N_T}+\frac{E_b}{N_0}\right)\right]$$

$$\cdot\ h\left(\frac{KE_b}{N_T}+\frac{KE_b}{N_0}\ ;\ m,2\right)$$

$$+\ \gamma^2 \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m+1}}{m+1} \exp\left[-\left(\frac{m}{m+1}\right)\cdot\left(\frac{KE_b}{N_T}\right)\right] h\left(\frac{KE_b}{N_T}\ ;\ m,2\right), \tag{4-34a}$$

where

$$h(x;m,2) = \frac{1}{m+1} \sum_{r=0}^{m} \binom{m}{r} \frac{r!}{(m+1)^r}\ \mathcal{L}_r^1\left(-\frac{x}{m+1}\right). \tag{4-34b}$$

Keeping in mind that $N_T = N_0 + N_J/\gamma$, we observe that maximization of (4-34) with respect to γ, which yields the worst-case partial-band performance, will involve a tradeoff between the magnitudes of the three terms and their weights. We also observe that such a maximization must be performed numerically because of the complexity of the expression.

4.2.4.4 No Thermal Noise

For $E_b/N_0 \to \infty$, the error probability vanishes except for the ℓ=L term:

$$P_b\left(e;\ \frac{E_b}{N_0} \to \infty\right) = \gamma^L \cdot \frac{1}{2}\,\frac{M}{M-1} \sum_{m=1}^{M-1} \binom{M-1}{m} \frac{(-1)^{m+1}}{m+1} \exp\left(-\frac{m}{m+1} \cdot \gamma R\right) h(\gamma R;\ m,L), \tag{4-35}$$

where $R \triangleq KE_b/N_J$. Unlike the M=2 case discussed in [1], for which it was possible to find an analytical solution for the worst case, for the M-ary situation the solution must be obtained numerically. However, it can be observed that maximization of (4-35) with respect to γ is equivalent to maximization of R^{-L} times (4-35) with respect to $X \equiv \gamma R$. Hence, for the special case of no thermal noise, $\gamma_0 = \text{const}/R$ and $P_b(e;\gamma_0) = \text{const}/R^L$, where the constants are functions of M and L.

4.3 NUMERICAL RESULTS FOR THE SQUARE-LAW AGC RECEIVER

In the following figures, the performance of the AGC receiver is summarized graphically, both for wideband noise jamming (γ=1) and for optimum partial-band noise jamming. The performance in optimum worst-case partial-band jamming was determined numerically by varying the partial-band fraction γ to find the maximum probability of bit error for given M, L, E_b/N_0, and E_b/N_J:

$$\text{worst case } P_b\left(e;\ \gamma,\ M,\ L,\ E_b/N_0,\ E_b/N_J\right)$$

$$= \max_{\gamma} P_b\left(e;\ \gamma,\ M,\ L,\ E_b/N_0,\ E_b/N_J\right). \tag{4-36}$$

This numerical procedure was followed because of the difficulty in obtaining an analytical solution to (4-36) by differentiating the error expression (4-31). The computer program for these calculations is given in Appendix 4E.

Comparison of systems with different values of M will be made on the following basis. For L=1, the systems corresponding to different values of M achieve the same bit error rate (BER) for different values of E_b/N_0. A BER of 10^{-n} is obtained for the values of E_b/N_0 given in Table 4-1. For most of the curves we will show, the baseline case will be a BER of 10^{-5} for each M; in effect we are comparing systems which are equivalent in performance under no jamming and prior to introduction of any multiple hopping. The variations in performance obtained by the systems for different M will be due to their different responses to increased L and to jamming effects.

TABLE 4-1

VALUES OF E_b/N_0 (IN dB) FOR WHICH BER = 1.0000×10^{-n}

M	n=3	n=5	n=7	n=9
2	10.94443	13.35247	14.89253	16.027135
4	8.35248	10.606572	12.07231	13.16444
8	6.971995	9.09401	10.49329	11.54624
16	6.069646	8.07835	9.41818	10.43496
32	5.418446	7.329656	8.61624	9.599615

4.3.1 Wideband Jamming (γ=1)

Having selected values of E_b/N_0 for the different M which yield BERs of 10^{-5}, we now consider the effects of wideband jamming. Because of the jamming, the SNR is no longer E_b/N_0, but

$$\frac{E_b}{N_T} = \frac{E_b}{N_0 + N_J} = \frac{\left(\frac{E_b}{N_0}\right)\left(\frac{E_b}{N_J}\right)}{\frac{E_b}{N_0} + \frac{E_b}{N_J}} \leq \frac{E_b}{N_0} . \tag{4-37}$$

Quite simply, for fixed E_b/N_0 the effect of wideband jamming is to reduce the effective SNR. Figure 4-2 illustrates for M=4 the fact that for wideband jamming the behavior of the BER for different L is the same as for no jamming. In this figure the BER is plotted against E_b/N_J, so that for low E_b/N_J the values are close to those for low E_b/N_0 in Figure 2-3; for high E_b/N_J, the BER of course approaches the value for the given $E_b/N_0 = 10.61$ dB, that is, 10^{-5} for L=1 and higher for $L > 1$ due to the noncoherent combining loss (NCL) effect.

A summary of wideband jamming results is given in Figure 4-3. The four complete curves represent the range of the parameters considered in the numerical computations: L (2 to 6) and M (2 to 32). Also, parts of two L=1 curves are shown to draw attention to the fact that for L=1 the M=2 and M=32 curves do not cross, whereas for $L > 1$ they do. This interesting behavior is due to the convention we have adopted for comparing the system performances for different M. For very high E_b/N_J, the performance for L=2 and M=32 is worse than for L=2 and M=2, due to a higher NCL at BER = 10^{-5}. As E_b/N_J decreases, the total SNR, E_b/N_T, decreases faster for M=2 than for M=32 and a crossover is experienced at the E_b/N_T for which the NCL is equal for both M's. For L=1 the curves in Figure 4-3 do not cross because by definition there is no NCL.

4.3.2 Optimum Jamming Fraction

Figure 4-4 shows the typical behavior of the optimum value γ_0 of γ, the partial-band jamming fraction, as a function of L, the number of hops per symbol, for M=4. It is seen that, in general, the value of γ_0 is inversely proportional to E_b/N_J when E_b/N_J exceeds some value; otherwise it is equal to one. Thus the jammer's strategy is to utilize wideband jamming when the available jamming power is relatively strong, and to use partial-band

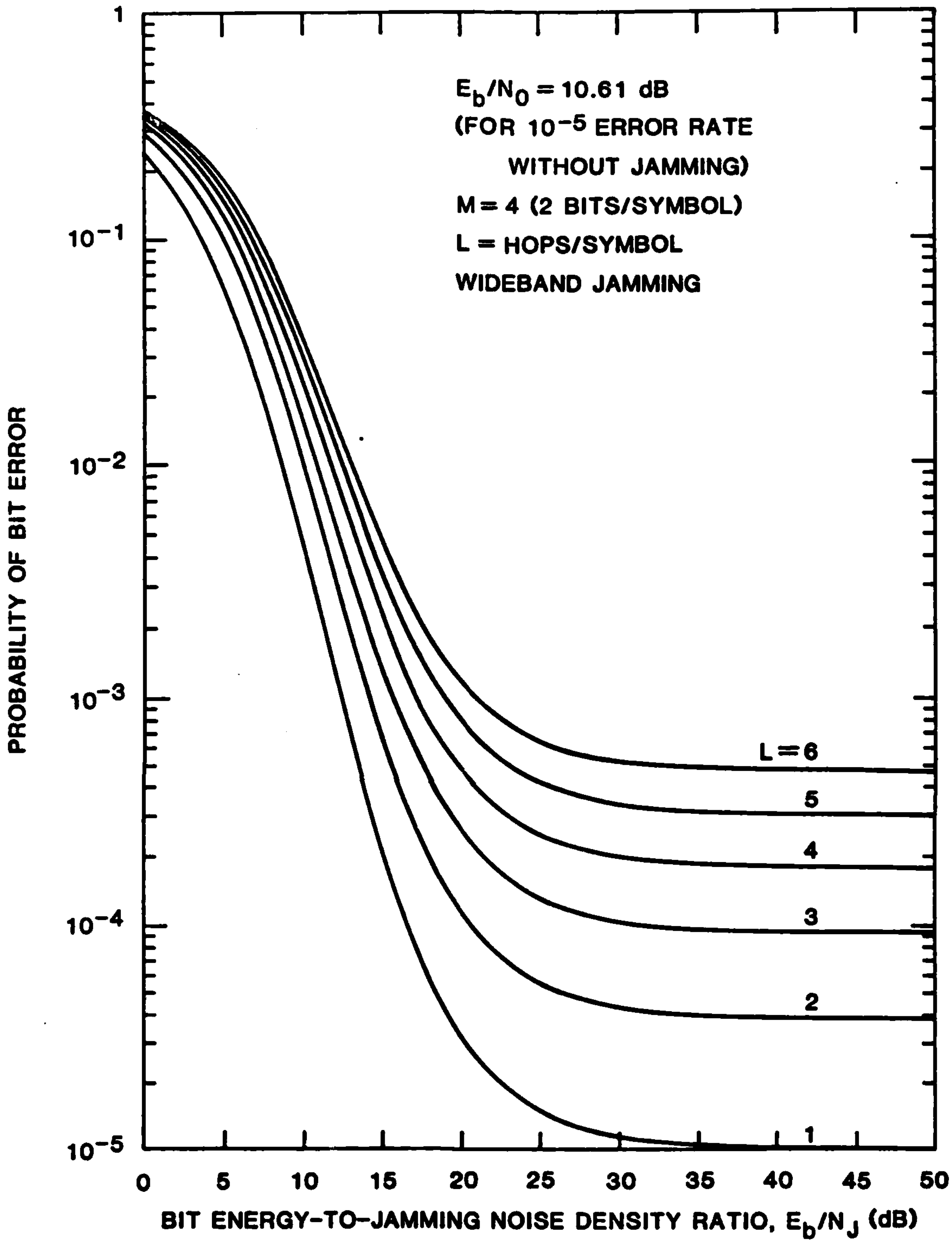


FIGURE 4-2 WIDEBAND JAMMING PERFORMANCE OF THE AGC FH/MFSK (M = 4) RECEIVER WHEN $E_b/N_0 = 10.61$ dB WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER

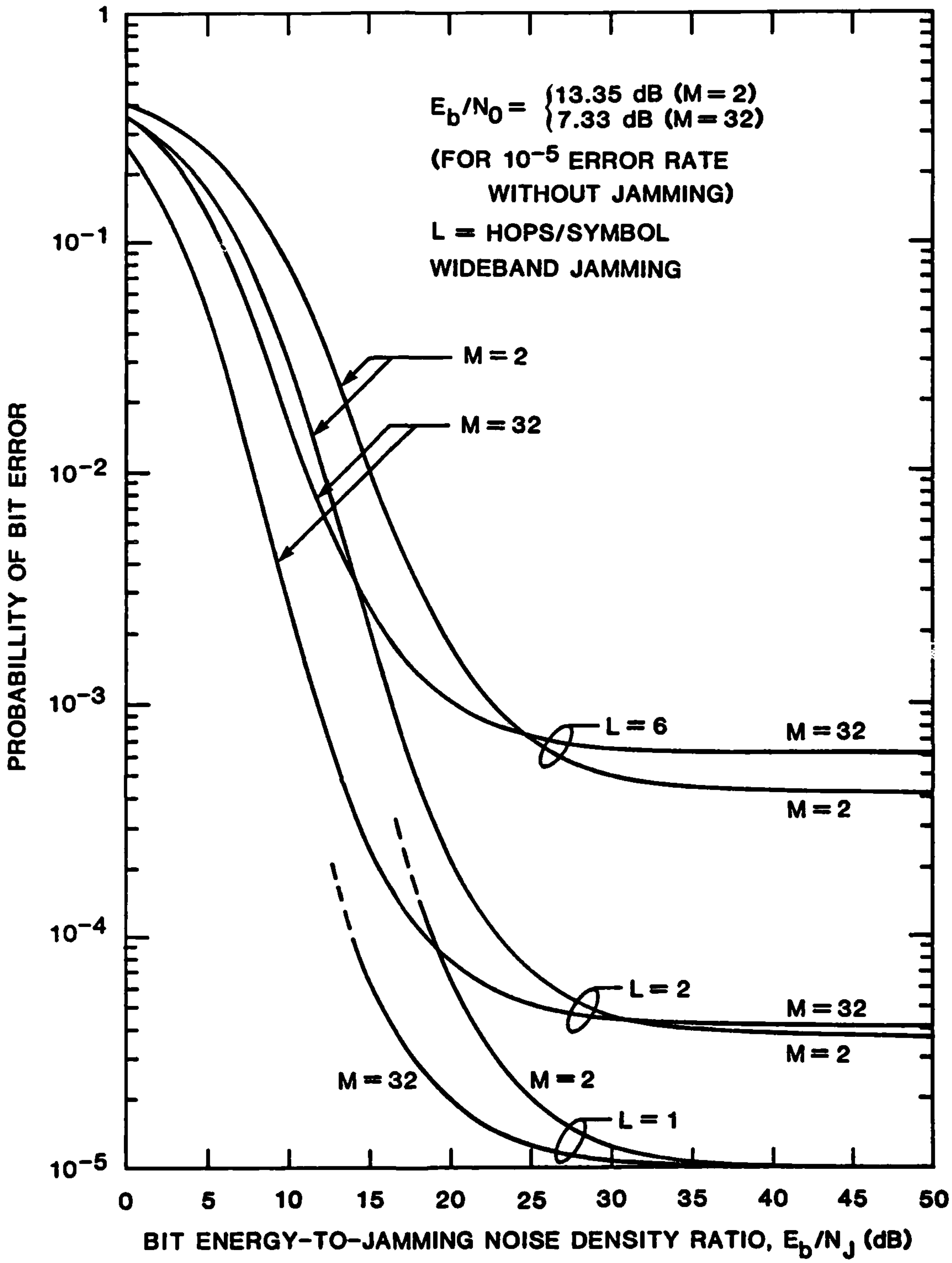


FIGURE 4-3 WIDEBAND JAMMING PERFORMANCE OF THE AGC RECEIVER FOR FH/MFSK WITH L = 1, 2, 6 AND M = 2, 32 WHEN E_b/N_0 = 13.35 dB (M = 2) AND E_b/N_0 = 7.33 dB (M = 32)

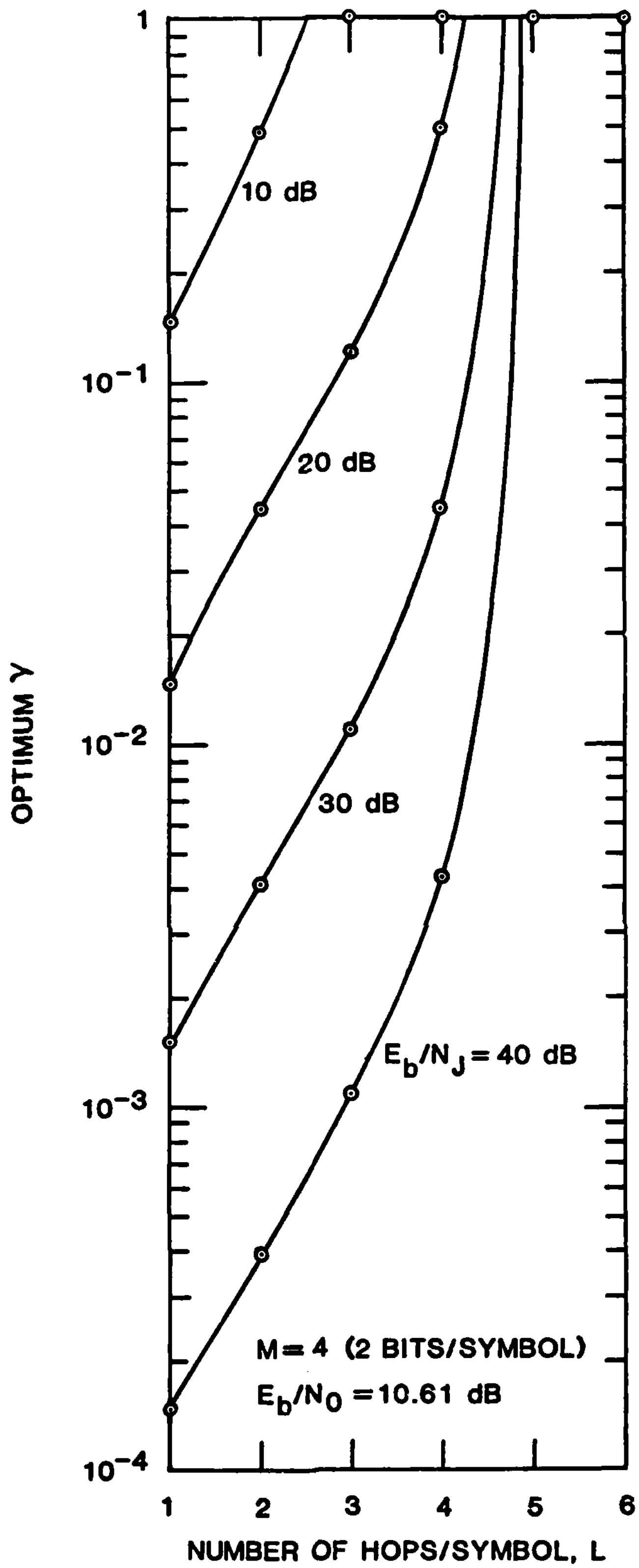


FIGURE 4-4 OPTIMUM JAMMING FRACTION (γ) VS. NUMBER OF HOPS/SYMBOL (L) FOR THE AGC FH/MFSK (M = 4) RECEIVER WHEN $E_b/N_0 = 10.61$ dB WITH E_b/N_J AS A PARAMETER (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

jamming when the jammer is weak compared to the signal. We observe that for the AGC receiver at the E_b/N_0 values chosen for the nominal case (BER = 10^{-5} and L=1), the value γ_0 is always unity for $L > 4$. The interpretation seems to be that for higher L the NCL acts as a form of self-jamming and therefore a wideband jammer can be effective against the system.

The above interpretation is supported by Figure 4-5, in which the M=4 case shown in Figure 4-4 is extended by increasing E_b/N_0 from 10.61 dB (BER = 10^{-5} for no jamming) to 13.16 dB (BER = 10^{-9} for no jamming). The effect of higher E_b/N_0 on the optimum γ is seen to be to lower it; the system at this value of E_b/N_0 has smaller NCL and thus the jammer must employ partial-band jamming to be most effective.

It may be observed that γ_0 is inversely proportional to M; evidence of this fact is given in Figure 4-6, which is a comparison of γ_0 vs. L curves for M=2 and M=32. From the γ_0 curves it is apparent that the jammer must acquire knowledge of M, L, and E_b/N_0 in order to select an effective partial-band jamming strategy.

4.3.3 Worst-Case Jamming Performance

The difference between wideband (γ=1) and optimum or worst-case partial-band noise jamming ($\gamma=\gamma_0$) effects on the AGC FH/MFSK receiver is most pronounced for L=1 hop/symbol, as illustrated in Figure 4-7 for M=8 and in Figure 4-8 for M=16. In these figures it is apparent that a weak jammer can increase the error rate by up to two orders of magnitude by employing partial-band jamming rather than wideband jamming. The resulting dependency of the BER upon E_b/N_J is approximately inverse linear, that is,

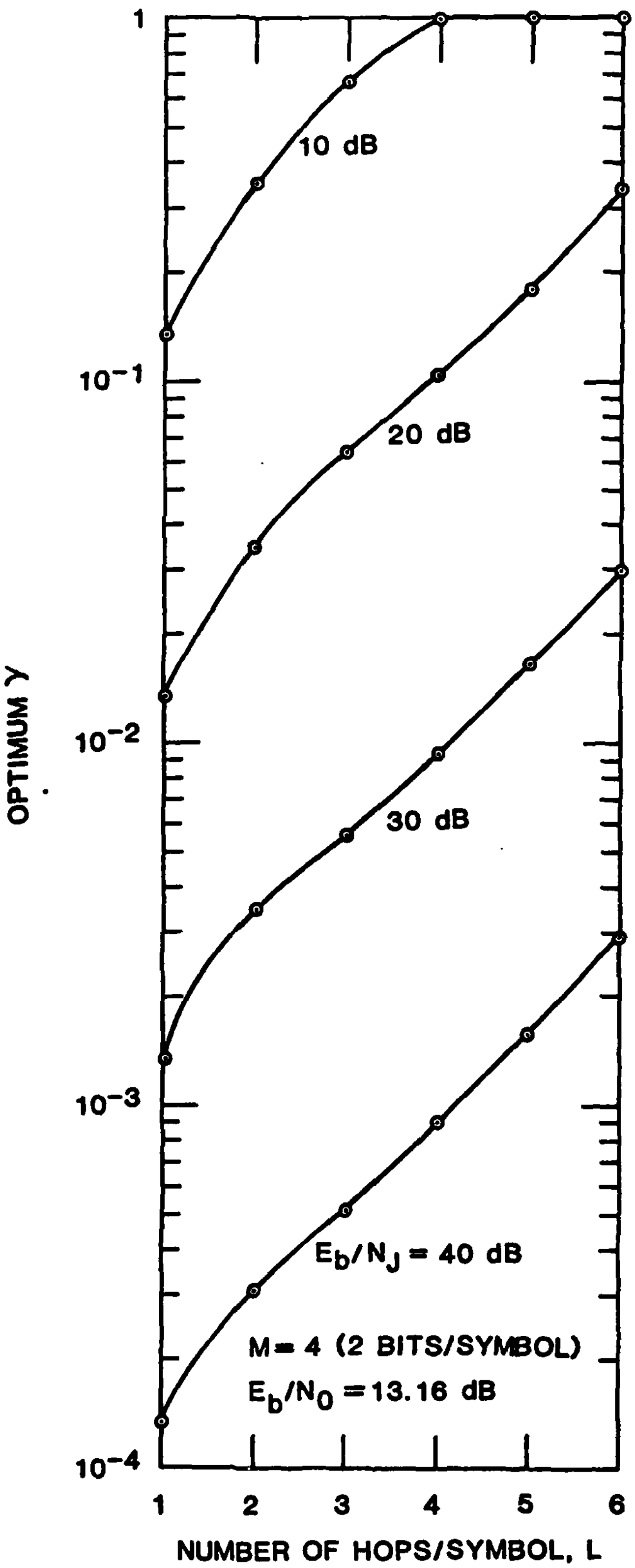


FIGURE 4-5 OPTIMUM JAMMING FRACTION (γ) VS. NUMBER OF HOPS/SYMBOL (L) FOR THE AGC FH/MFSK (M = 4) RECEIVER WHEN E_b/N_0 = 13.16 dB WITH E_b/N_J AS A PARAMETER (FOR 10^{-9} ERROR RATE WITHOUT JAMMING)

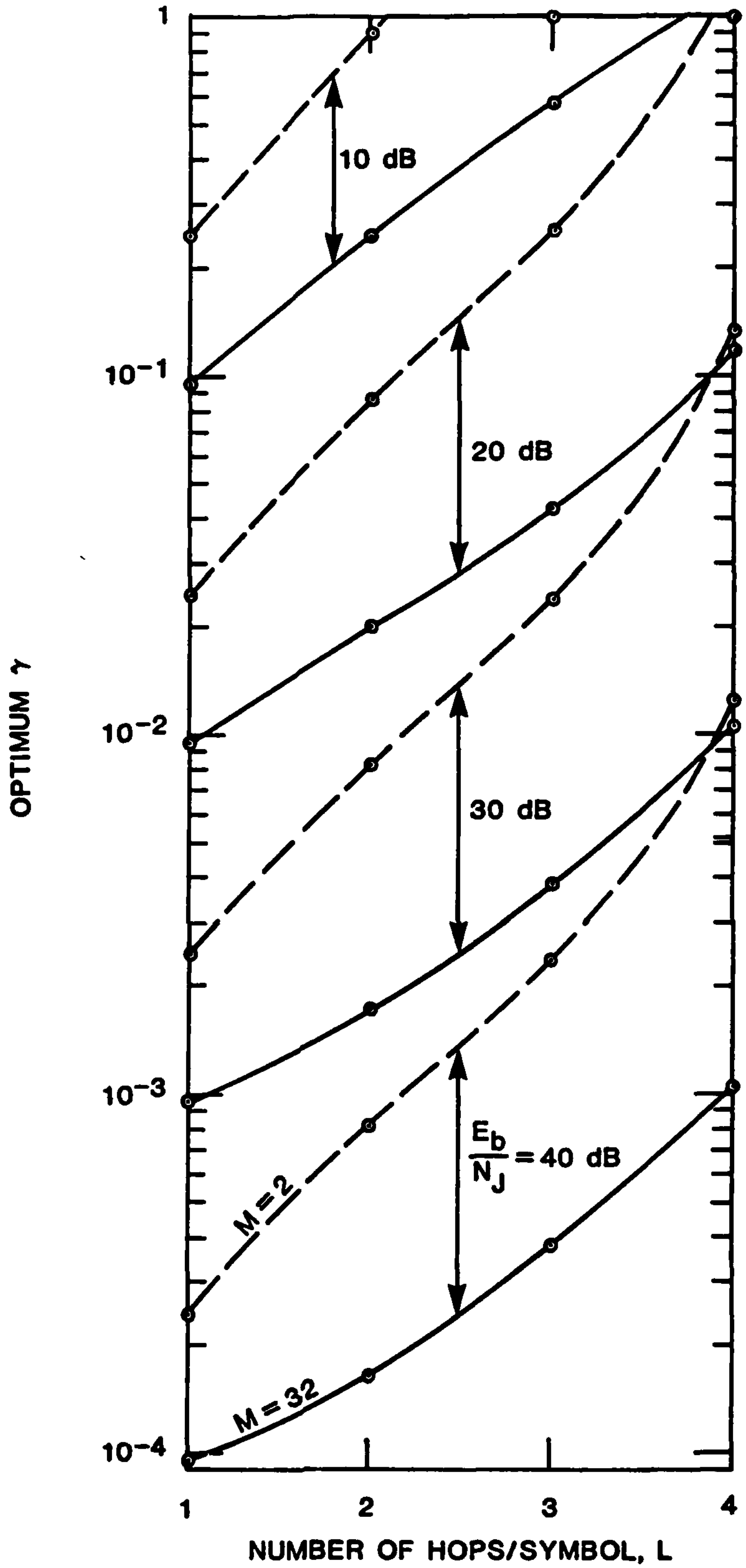


FIGURE 4-6 OPTIMUM JAMMING FRACTION (γ) VS. NUMBER OF HOPS/SYMBOL (L) FOR THE AGC FH/MFSK RECEIVER WITH E_b/N_J AS A PARAMETER WHEN M = 2 (E_b/N_0 = 13.35 dB) AND M = 32 (E_b/N_0 = 7.33 dB) (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

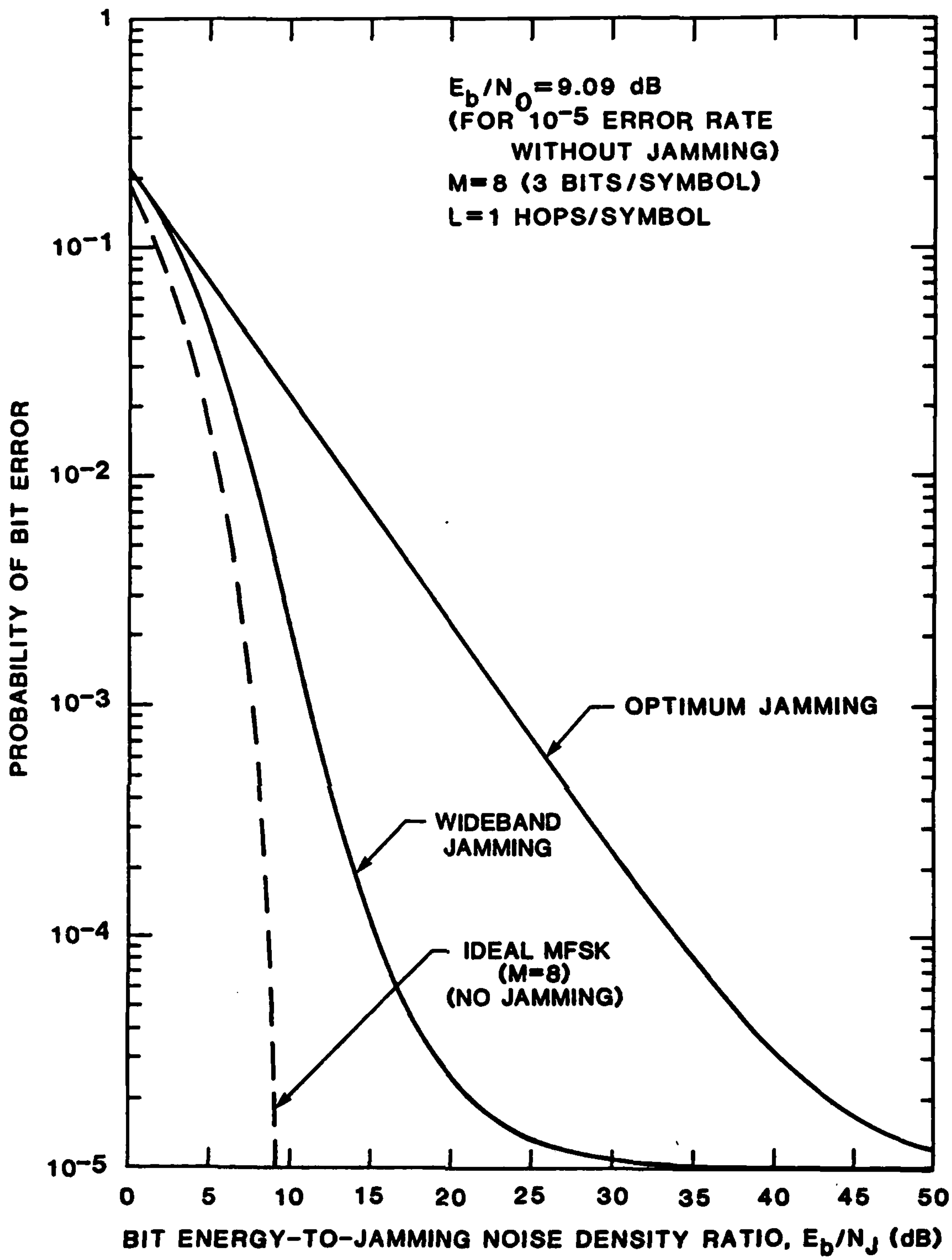


FIGURE 4-7 OPTIMUM PARTIAL-BAND JAMMING AND WIDEBAND JAMMING PERFORMANCES OF AGC FH/MFSK (M=8) RECEIVER FOR L=1 HOP/SYMBOL WHEN E_b/N_0=9.09 dB (FOR IDEAL MFSK (M=8) CURVE THE ABSCISSA READS E_b/N_0)

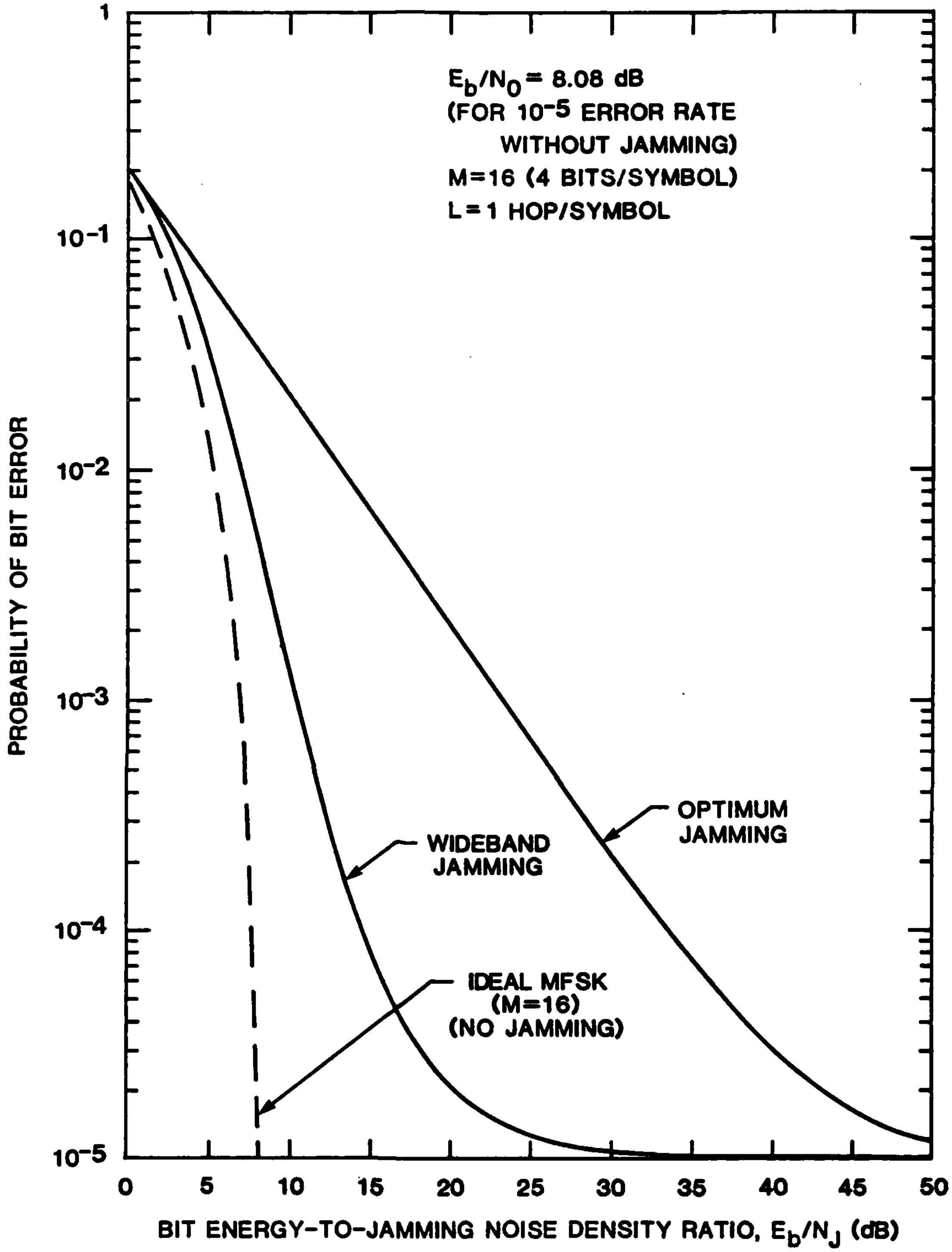


FIGURE 4-8 OPTIMUM PARTIAL-BAND JAMMING AND WIDEBAND JAMMING PERFORMANCES OF AGC FH/MFSK (M=16) RECEIVER FOR L=1 HOP/SYMBOL WHEN E_b/N_0=8.08 dB (FOR IDEAL MFSK (M=16) CURVE THE ABSCISSA READS E_b/N_0)

$$\max_{\gamma} P_b(e;\gamma,\ L=1) \approx \text{const} \Big/ \frac{E_b}{N_J}\ , \tag{4-38}$$

for E_b/N_J between 0 dB and 40 dB. For extremely high E_b/N_J, of course, the BER is determined by E_b/N_0, which was chosen to be the value which makes the error rate equal to 10^{-5}.

As L is increased from the value L=1, two effects occur, as shown for L=2 in Figure 4-9 for M=8 and in Figure 4-10 for M=16. First, the wide-band jamming performance is pushed up or degraded due to the noncoherent combining loss effect, so that the BER for high E_b/N_J approaches 4.1×10^{-5} for L=1. Second, the optimum jamming performance is improved greatly; the largest difference in performance is about a factor of three rather than two orders of magnitude. This indicates that the normalization employed by the AGC receiver is successful in combatting the partial-band jamming, which after dehopping appears to the receiver to be a kind of pulsed or intermittent jamming. The normalization in effect weighs the jammed hops less in the symbol decision, countering the tendency of the jamming to obscure the difference in average power between the signal-plus-noise channel and the noise-only channels. This can be seen by considering the ratio of the average powers of the signal and noise-only channels for ℓ hops jammed:

$$\frac{E(z_1)}{E(z_2)} = \frac{2L + 2\rho_\ell}{2L} = 1 + \frac{L-\ell}{L} \cdot \frac{S}{\sigma_N^2} + \frac{\ell}{L} \cdot \frac{S}{\sigma_T^2} \tag{4-39a}$$

for the AGC receiver, and

$$\frac{E(z_1)}{E(z_2)} = \frac{2(L-\ell)\sigma_N^2 + 2\ell\sigma_T^2 + 2LS}{2(L-\ell)\sigma_N^2 + 2\ell\sigma_T^2} = 1 + \frac{S}{\frac{(L-\ell)}{L}\sigma_N^2 + \frac{\ell}{L}\sigma_T^2} \tag{4-39b}$$

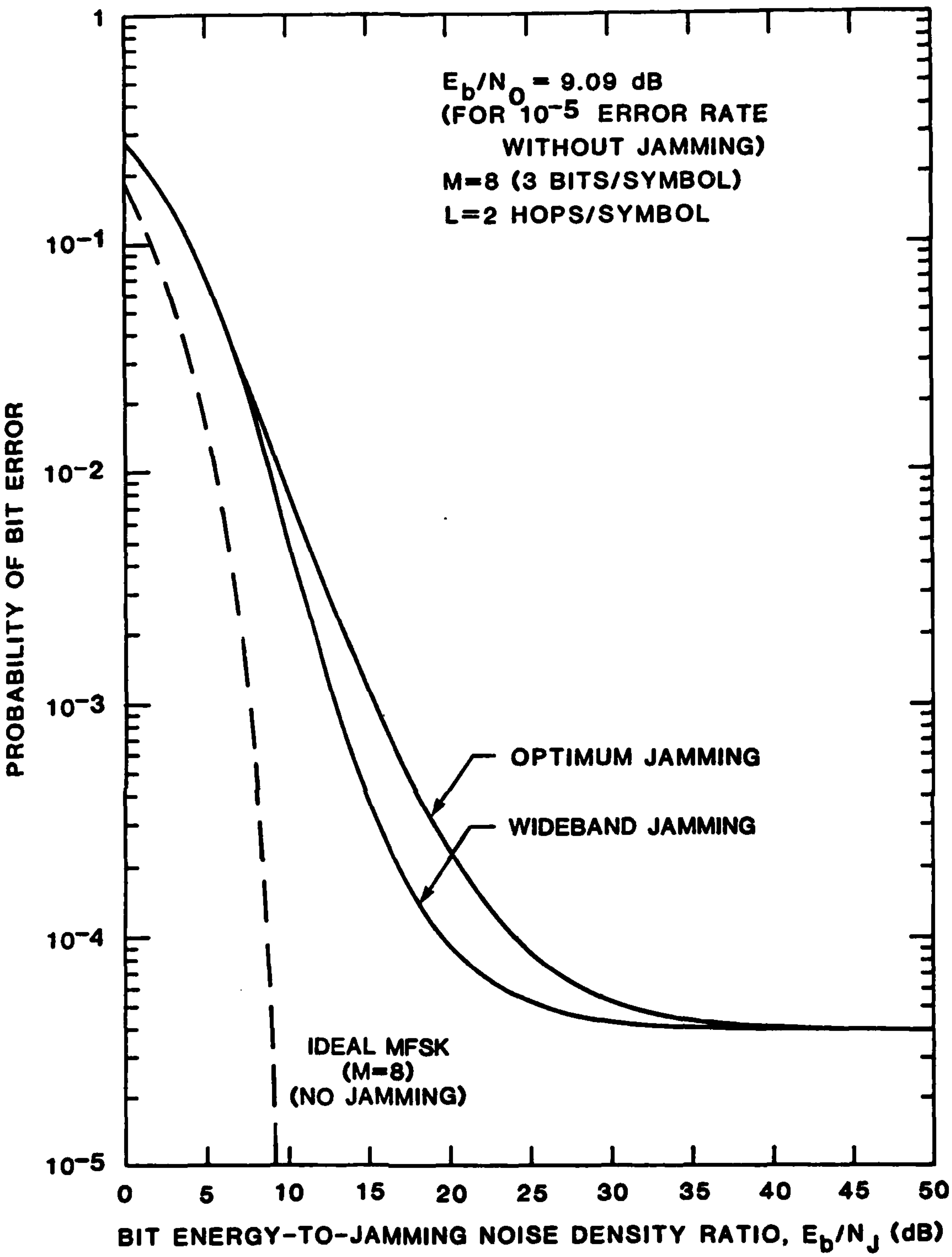


FIGURE 4-9 OPTIMUM PARTIAL-BAND JAMMING AND WIDEBAND JAMMING PERFORMANCES OF AGC FH/MFSK (M=8) RECEIVER FOR L=2 HOPS/SYMBOL WHEN E_b/N_0 = 9.09 dB (FOR IDEAL MFSK (M=8) CURVE THE ABSCISSA READS E_b/N_0)

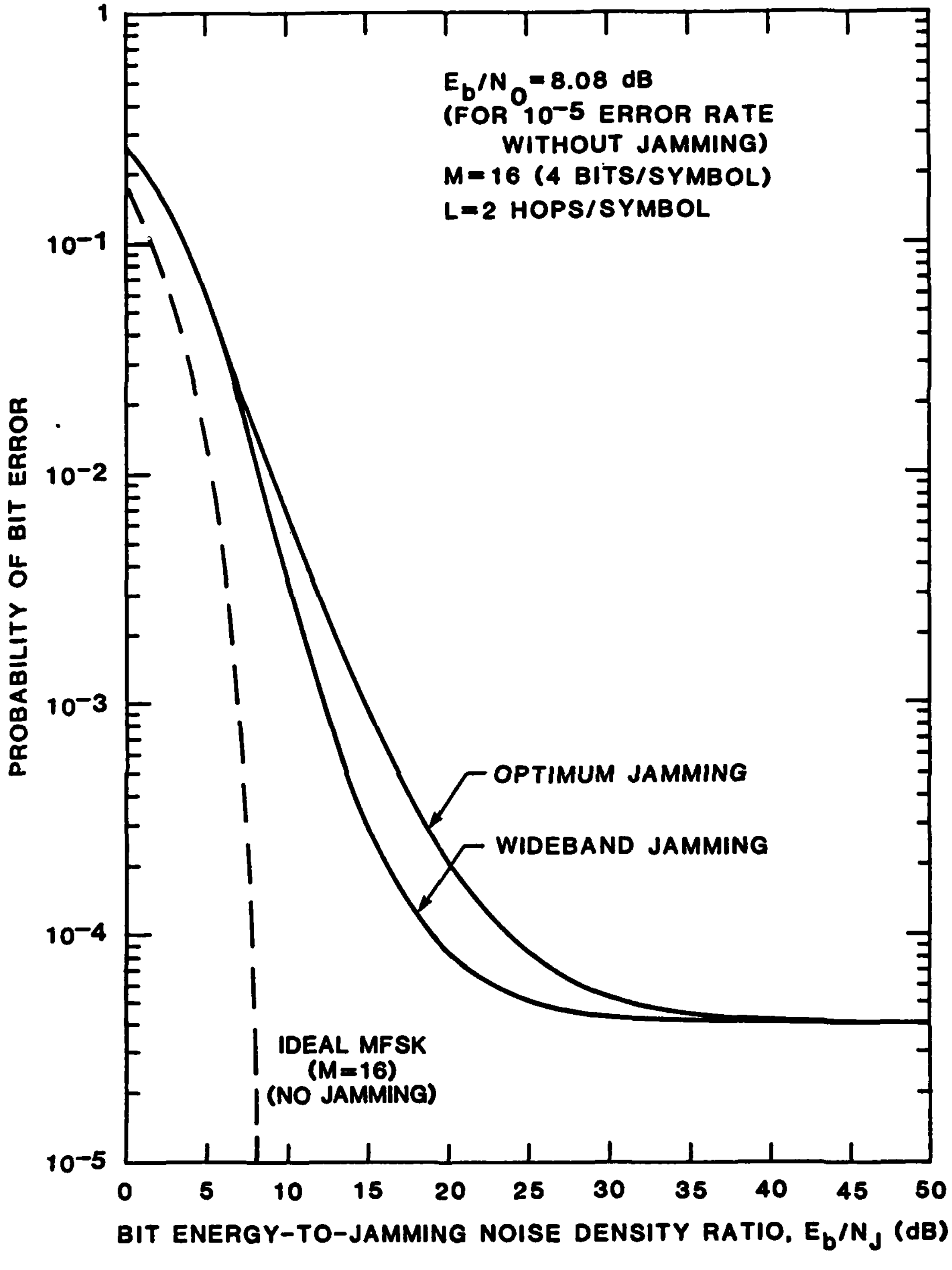


FIGURE 4-10 OPTIMUM PARTIAL-BAND JAMMING AND WIDEBAND JAMMING PERFORMANCES OF AGC FH/MFSK (M=16) RECEIVER FOR L=2 HOPS/SYMBOL WHEN E_b/N_0=8.08 dB (FOR IDEAL MFSK (M=16) CURVE THE ABSCISSA READS E_b/N_0)

for the conventional receiver. For strong jamming ($\sigma_T^2 \rightarrow \infty$), the AGC's action as represented by (4-39a) tends toward a value greater than one, while the conventional receiver as represented by (4-39b) tends toward one, or no difference in the average values of the decision statistics upon which the symbol decision is based.

The countering or anti-jam effectiveness of the AGC normalization improves as L increases, as illustrated for L=4 by Figure 4-11 for M=8 and by Figure 4-12 for M=16. This is consistent with the γ_0 behavior seen before (Figure 4-4) in which $\gamma_0 \rightarrow 1$ as $L \rightarrow 5$ for the BER=10^{-5} series of curves. As L increases the difference between optimum and wideband jamming effects becomes small; however, for high E_b/N_J the error rate increases because of the NCL effects. Thus there is a tradeoff between antijam capability and noncoherent combining losses.

The worst-case jamming performances of the AGC receiver for different values of M, the symbol alphabet size, are shown in Figures 4-13 to 4-16 for L = 2, 3, 4, and 6; these curves are also summarized by Figure 4-17. Since the average bit energy-to-noise density ratio is

$$\left(\frac{E_b}{N_0}\right)_{avg} = (1-\gamma_0)\frac{E_b}{N_0} + \gamma_0 \frac{E_b}{N_0 + N_J/\gamma_0} \approx \begin{cases} \dfrac{E_b}{N_J}, & \dfrac{E_b}{N_J} \text{ small } (\gamma_0 \simeq 1), \\[2ex] \dfrac{E_b}{N_0}, & \dfrac{E_b}{N_J} \text{ large}, \end{cases} \qquad (4\text{-}40)$$

for small E_b/N_J the results for different M reflect a bit-energy constraint comparison similar to Figure 2-2, that is, the error rate decreases as M increases. For large E_b/N_J, effectively there is no jamming, and the BER increases with M, as explained in Section 2.1.

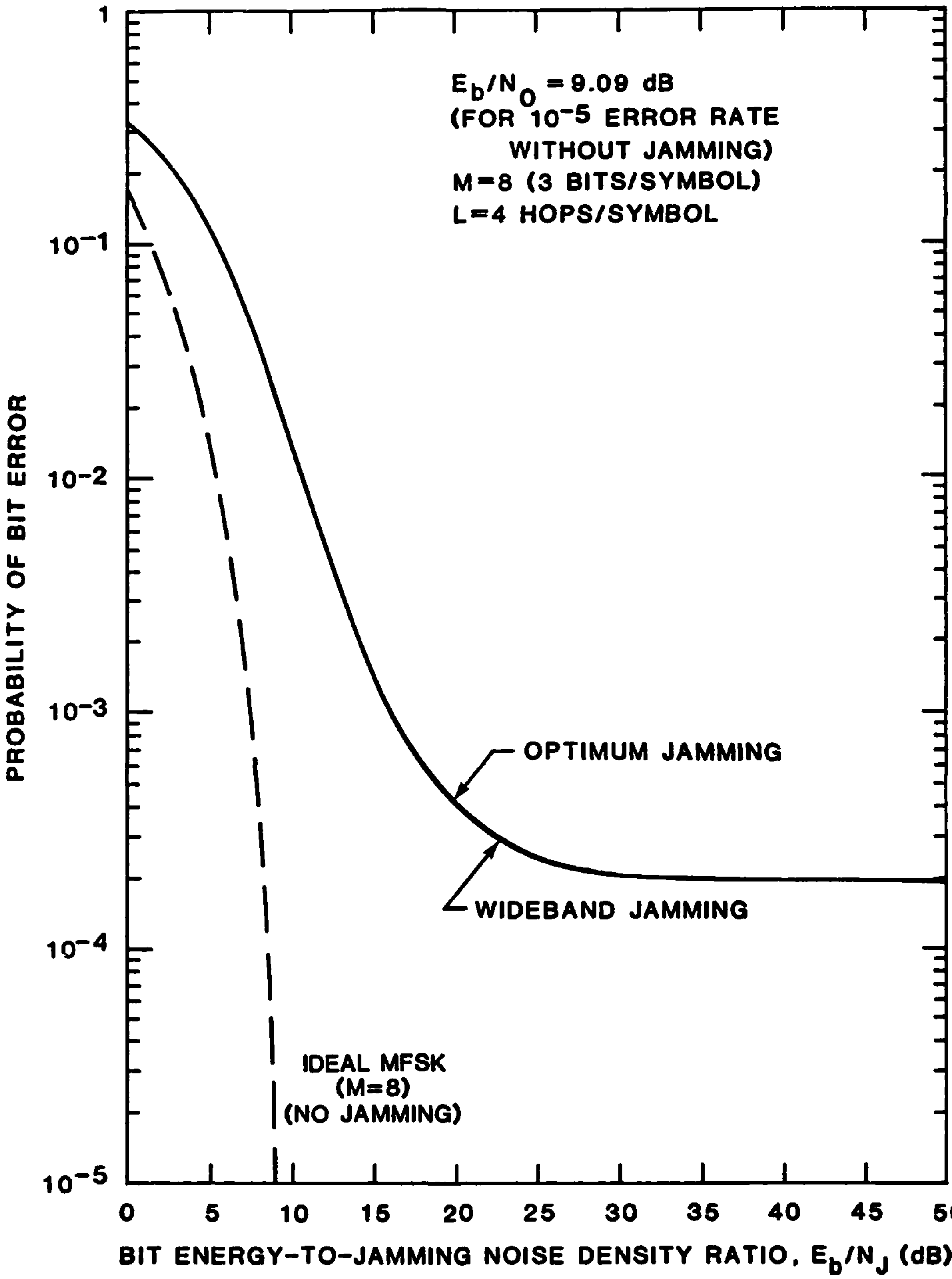


FIGURE 4-11 OPTIMUM PARTIAL-BAND JAMMING AND WIDEBAND JAMMING PERFORMANCES OF AGC FH/MFSK (M=8) RECEIVER FOR L=4 HOPS/SYMBOL WHEN $E_b/N_0 = 9.09$ dB (FOR IDEAL MFSK (M=8) CURVE THE ABSCISSA READS E_b/N_0)

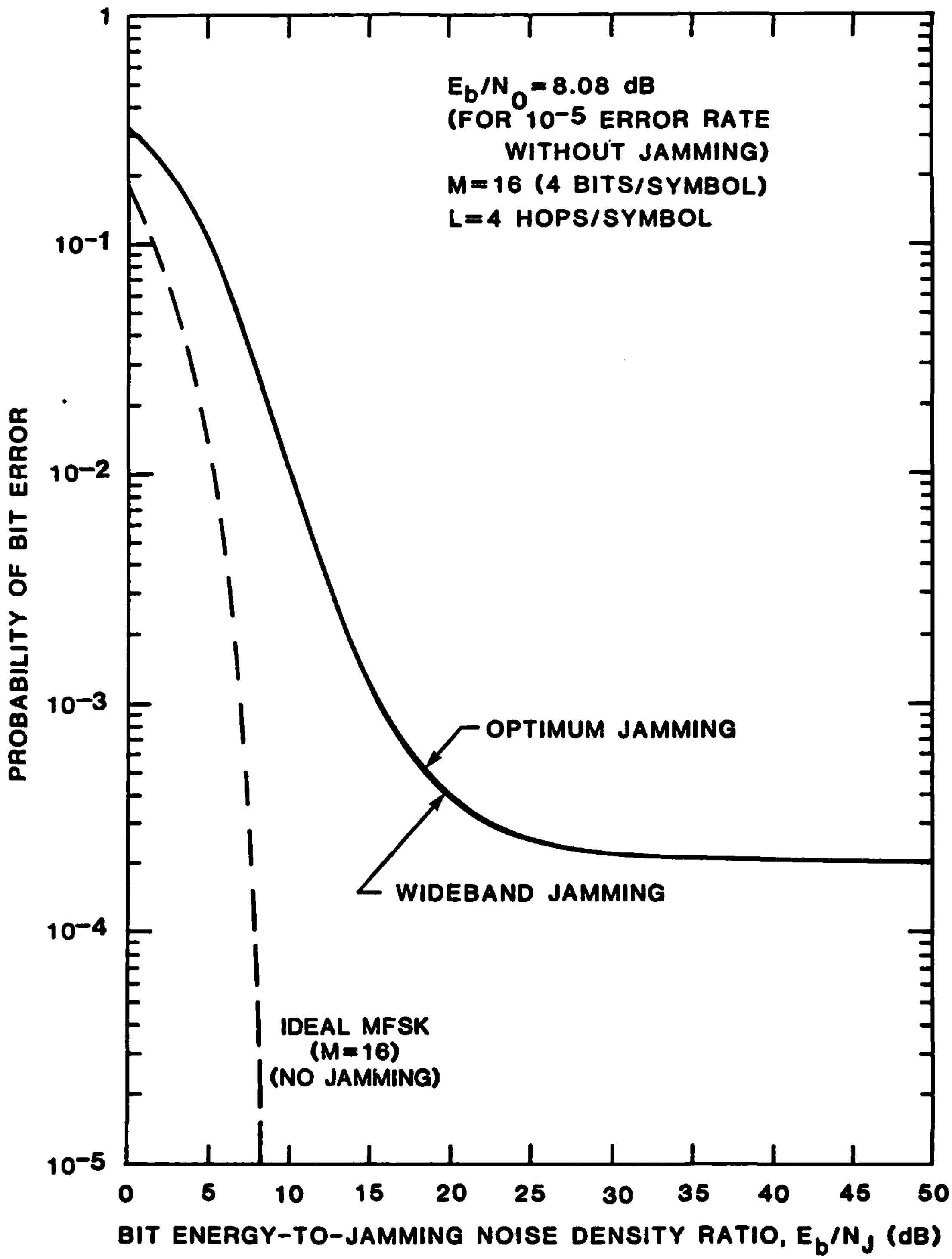


FIGURE 4-12 OPTIMUM PARTIAL-BAND JAMMING AND WIDEBAND JAMMING PERFORMANCES OF AGC FH/MFSK (M=16) RECEIVER FOR L=4 HOPS/SYMBOL WHEN $E_b/N_0 = 8.08$ dB (FOR IDEAL MFSK (M=16) CURVE THE ABSCISSA READS E_b/N_0)

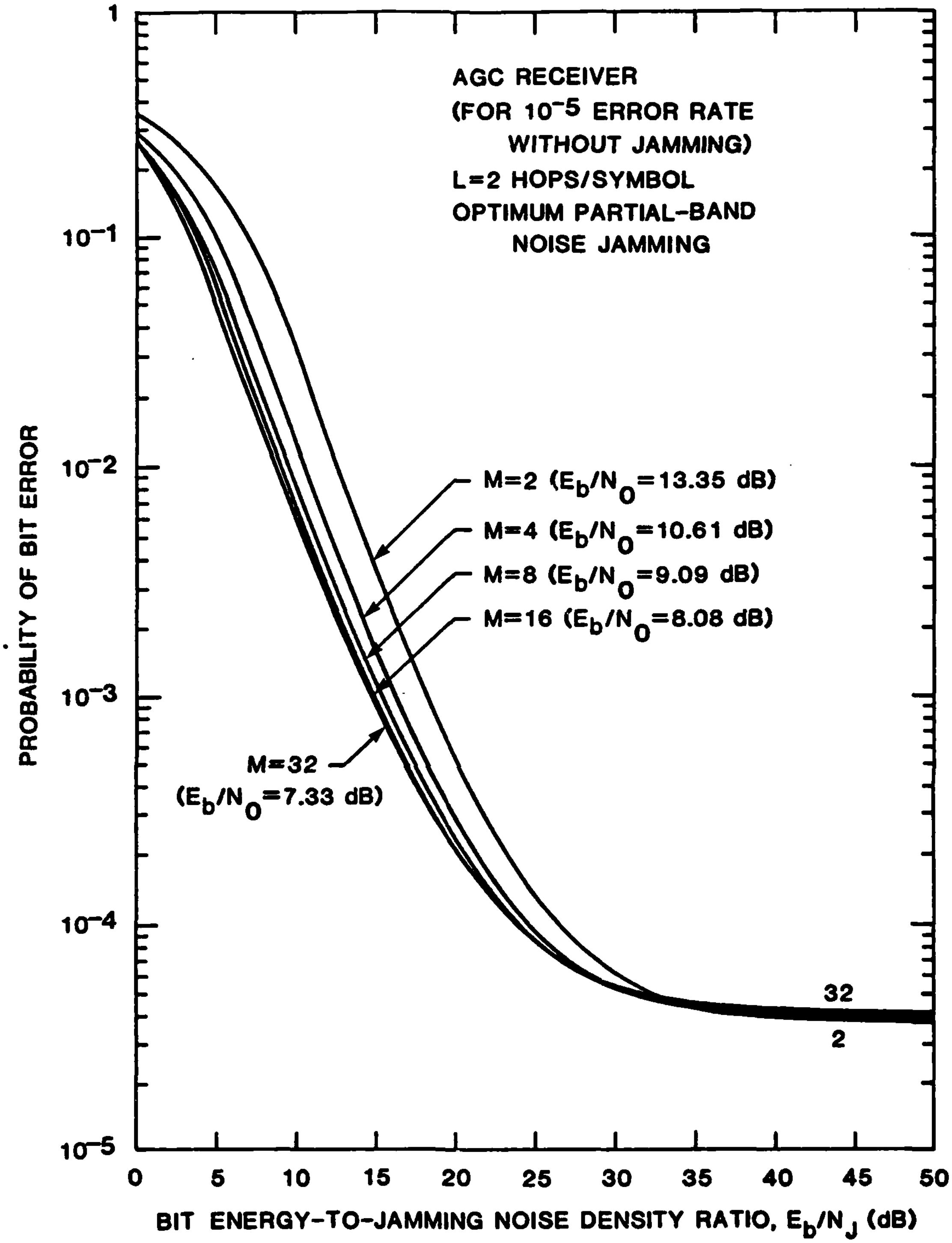


FIGURE 4-13 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF AGC FH/MFSK RECEIVER FOR L=2 HOPS/SYMBOL WITH THE NUMBER OF SYMBOLS (M) AS A PARAMETER

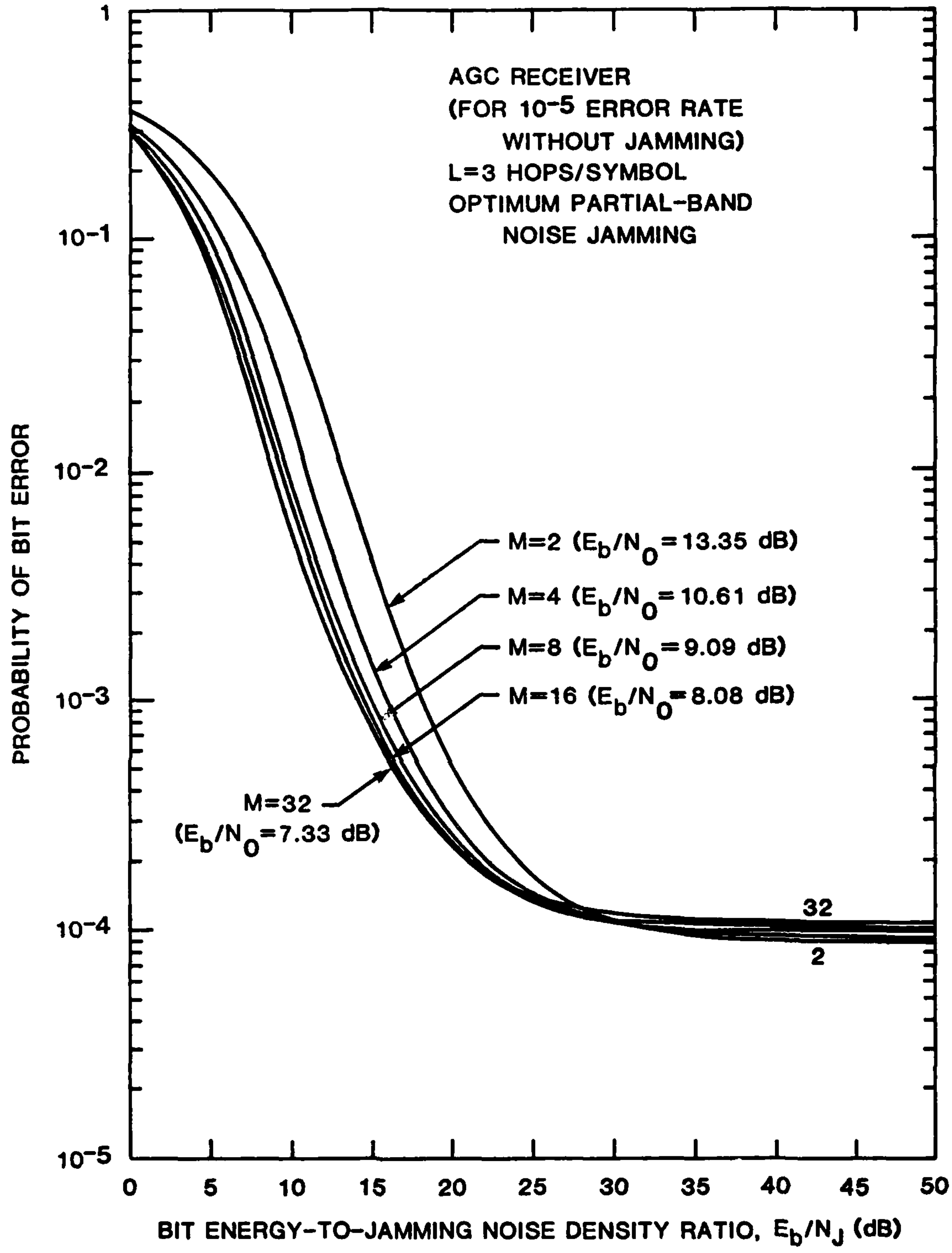
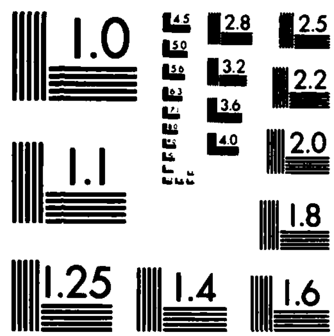


FIGURE 4-14 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF AGC FH/MFSK RECEIVER FOR L=3 HOPS/SYMBOL WITH THE NUMBER OF SYMBOLS (M) AS A PARAMETER

AD-A147 766 OPTIMUM JAMMING EFFECTS ON FREQUENCY-HOPPING M-ARY FSK 3/7
(FREQUENCY-SHIFT K..(U) LEE (J S) ASSOCIATES INC
ARLINGTON VA J S LEE ET AL. OCT 84 JC-2025-N
UNCLASSIFIED N00014-83-C-0312 F/G 17/4 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

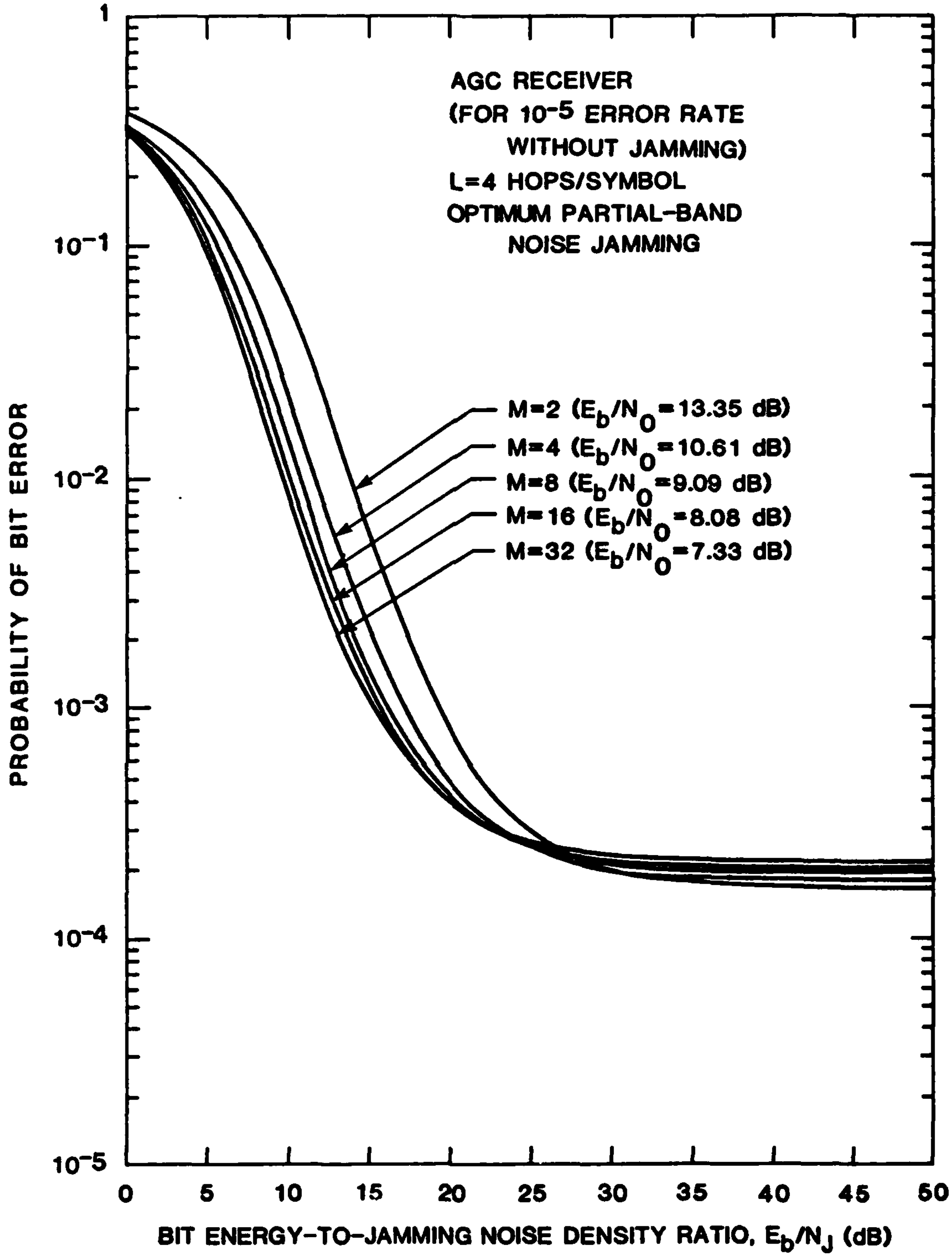


FIGURE 4-15 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF AGC FH/MFSK RECEIVER FOR L=4 HOPS/SYMBOL WITH THE NUMBER OF SYMBOLS (M) AS A PARAMETER

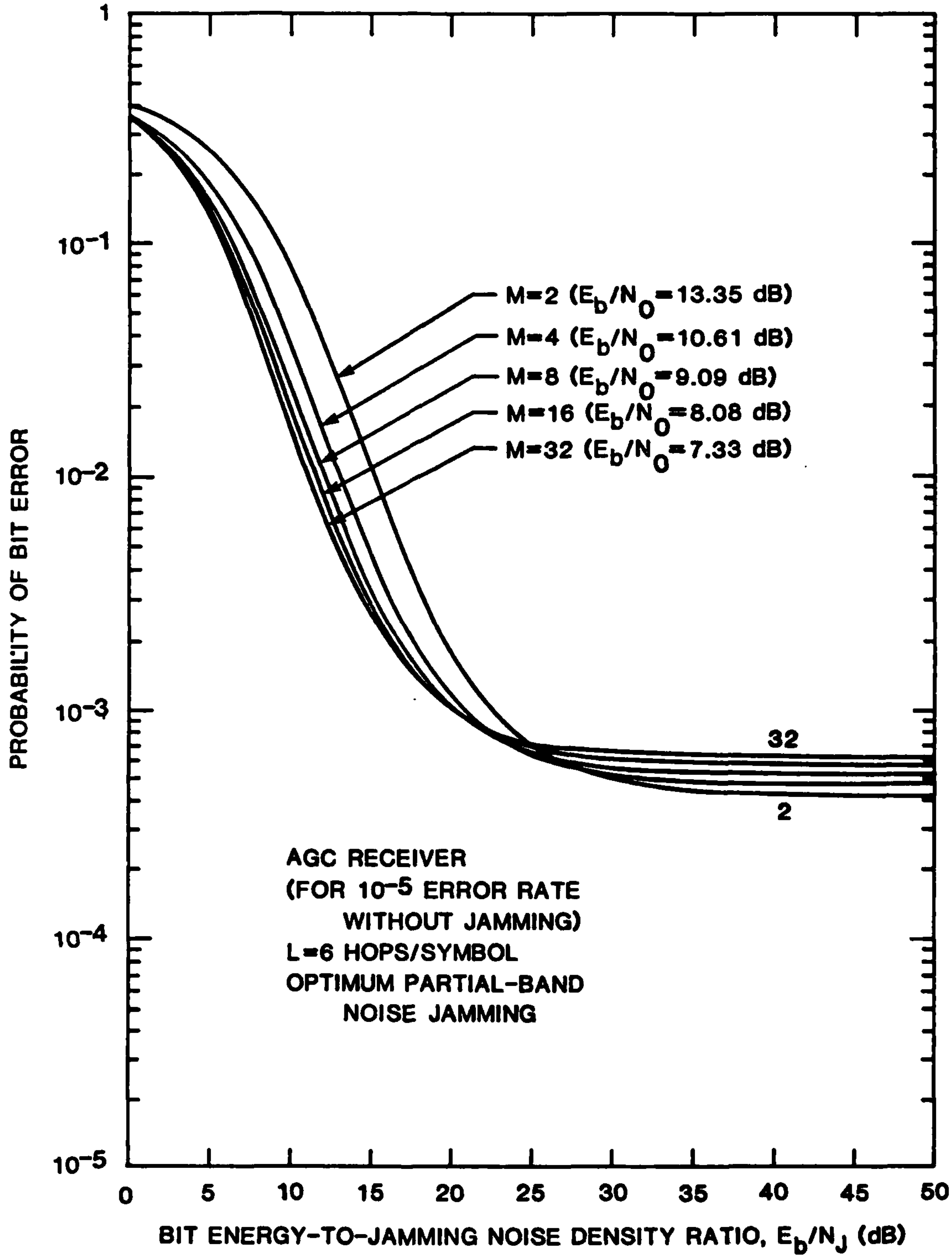


FIGURE 4-16 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF AGC FH/MFSK RECEIVER FOR L=6 HOPS/SYMBOL WITH THE NUMBER OF SYMBOLS (M) AS A PARAMETER

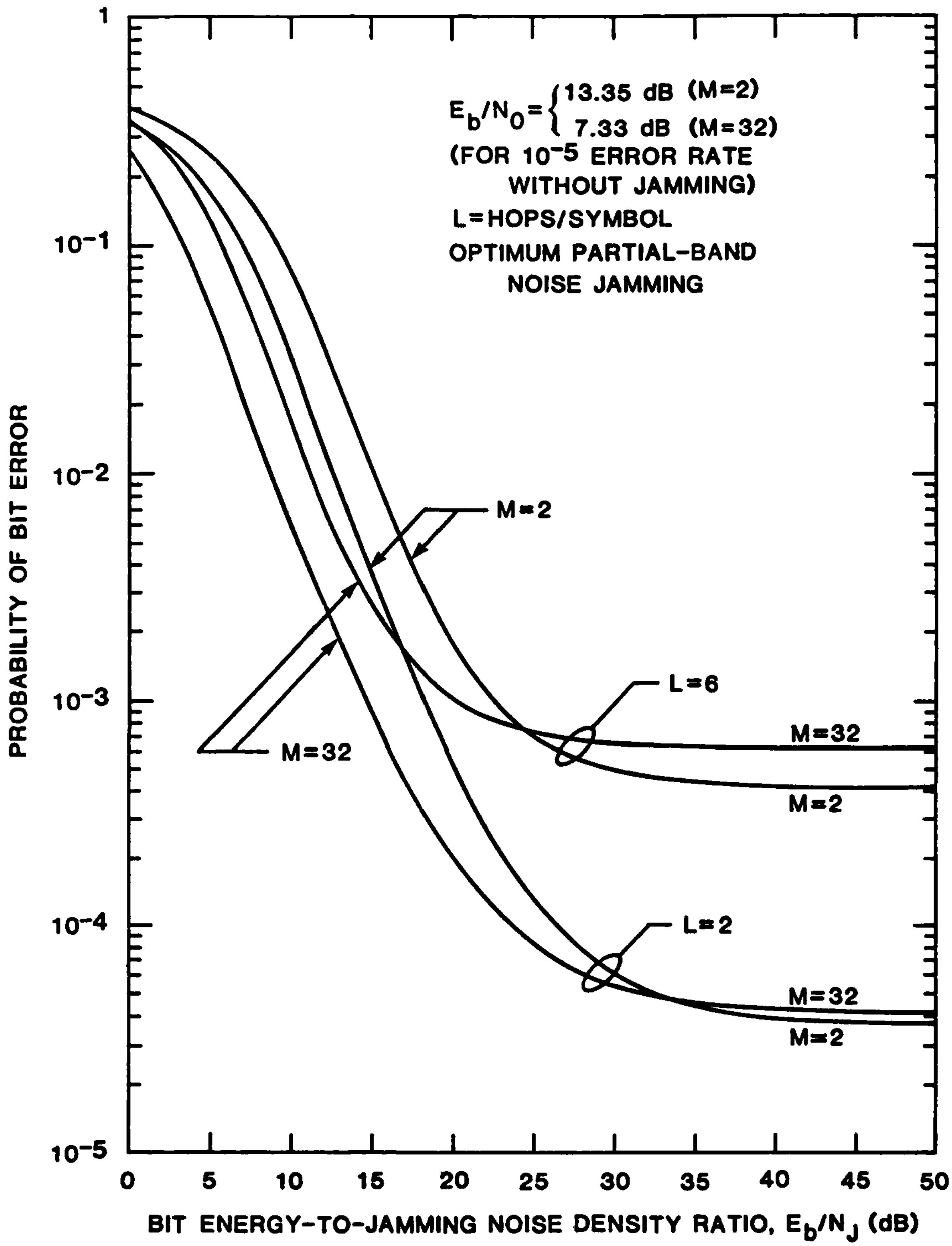


FIGURE 4-17 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF AGC FH/MFSK RECEIVER FOR M=2, 32 AND L=2, 6 HOPS/SYMBOL WHEN E_b/N_0=13.35 dB (M=2) AND 7.33 dB (M=32)

The tradeoff between antijam effectiveness and NCL which takes place as L, the number of hops per symbol, is increased is well illustrated in Figures 4-18 through 4-22. Each of these figures gives the bit error rate as a function of E_b/N_J for a given value of M and E_b/N_0, and for different values of L (L=1,2,3,4,6). By plotting the performances obtained for different L on the same graph we are able to observe that a kind of diversity improvement is obtained for E_b/N_J between approximately 5 dB and 40 dB. The improvement is a limited one, unlike the typical diversity improvements gained under fading conditions. For the particular values of E_b/N_0 used in these figures, the performances for L=2 and L=3 are better than that for L=1 at certain values of E_b/N_J, but for small or very large E_b/N_J, the L=1 system is best.

Identification of NCL as the limiting factor in diversity improvement with increasing L is confirmed in Figures 4-23 to 4-25. In Figure 4-23, for M=2, E_b/N_0 is increased such that the error rate is 10^{-7} for L=1 and no jamming; the limitation of the improvement occurs later, that is at a higher E_b/N_J and lower BER, than in Figure 4-18, so that the optimum diversity gets to be as high as L=4 for a small range of E_b/N_J.

In Figure 4-24, for M=4, E_b/N_0 is again increased, such that the BER = 10^{-9} for L=1 and no jamming. In this case the highest optimum diversity is L=5, for a small range of E_b/N_J beginning around E_b/N_J = 18 dB.

As $E_b/N_0 \to \infty$, we have the case of no thermal noise. Figure 4-25 shows, for the example of M=2, that the system performance improves indefinitely with increasing E_b/N_J and that the optimum diversity increases also. If the appropriate value of L is chosen the performance obtained is within 3 dB of the ideal system performance for L=1. However, this situation is very idealistic, and we anticipate that as a compromise a realistic system might employ L=2, since most of the improvement over L=1 is obtained for this case.

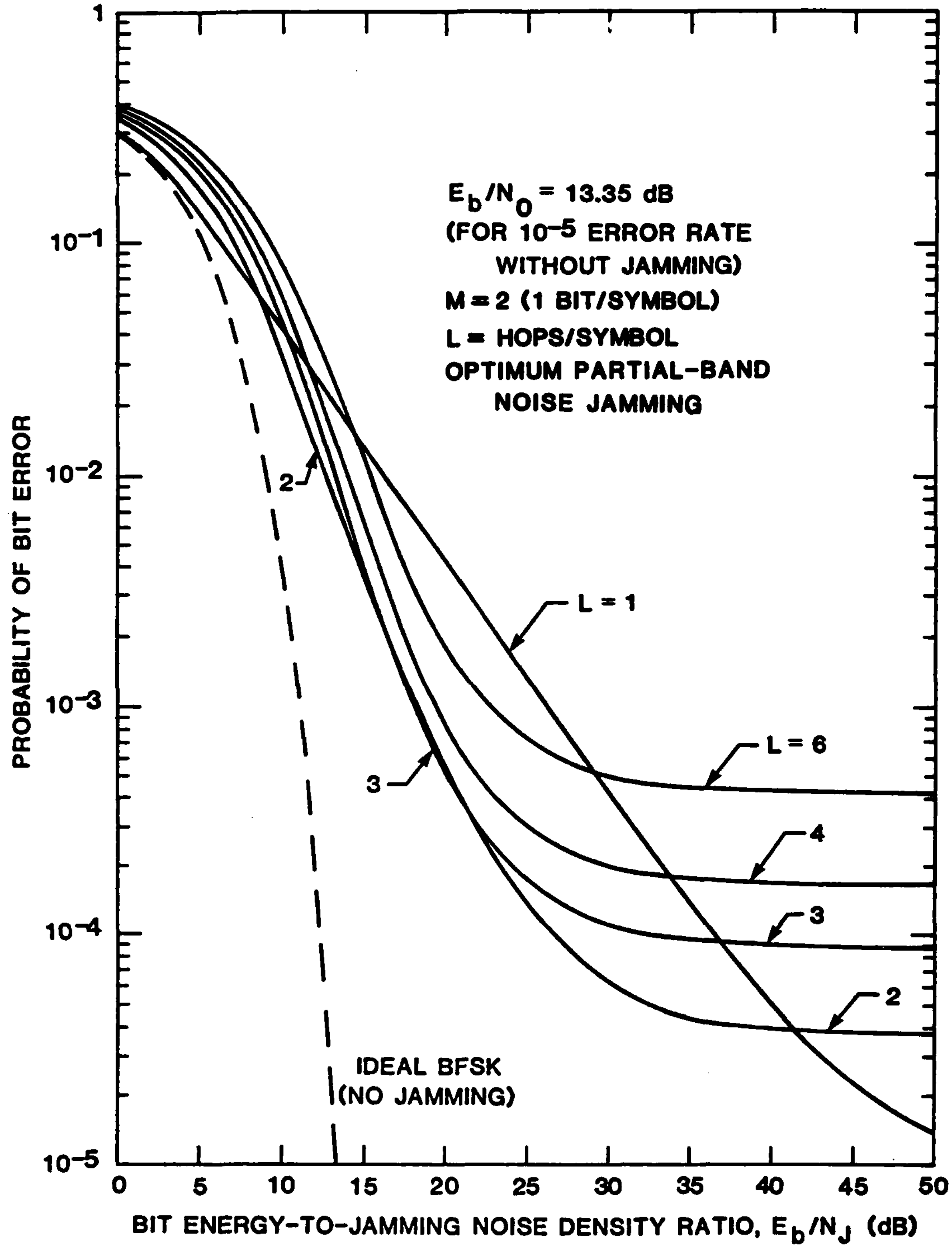


FIGURE 4-18 OPTIMUM JAMMING PERFORMANCE OF THE AGC RECEIVER FOR BFSK/FH WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER WHEN E_b/N_0 = 13.35 dB (FOR IDEAL BFSK CURVE THE ABSCISSA READS E_b/N_0)

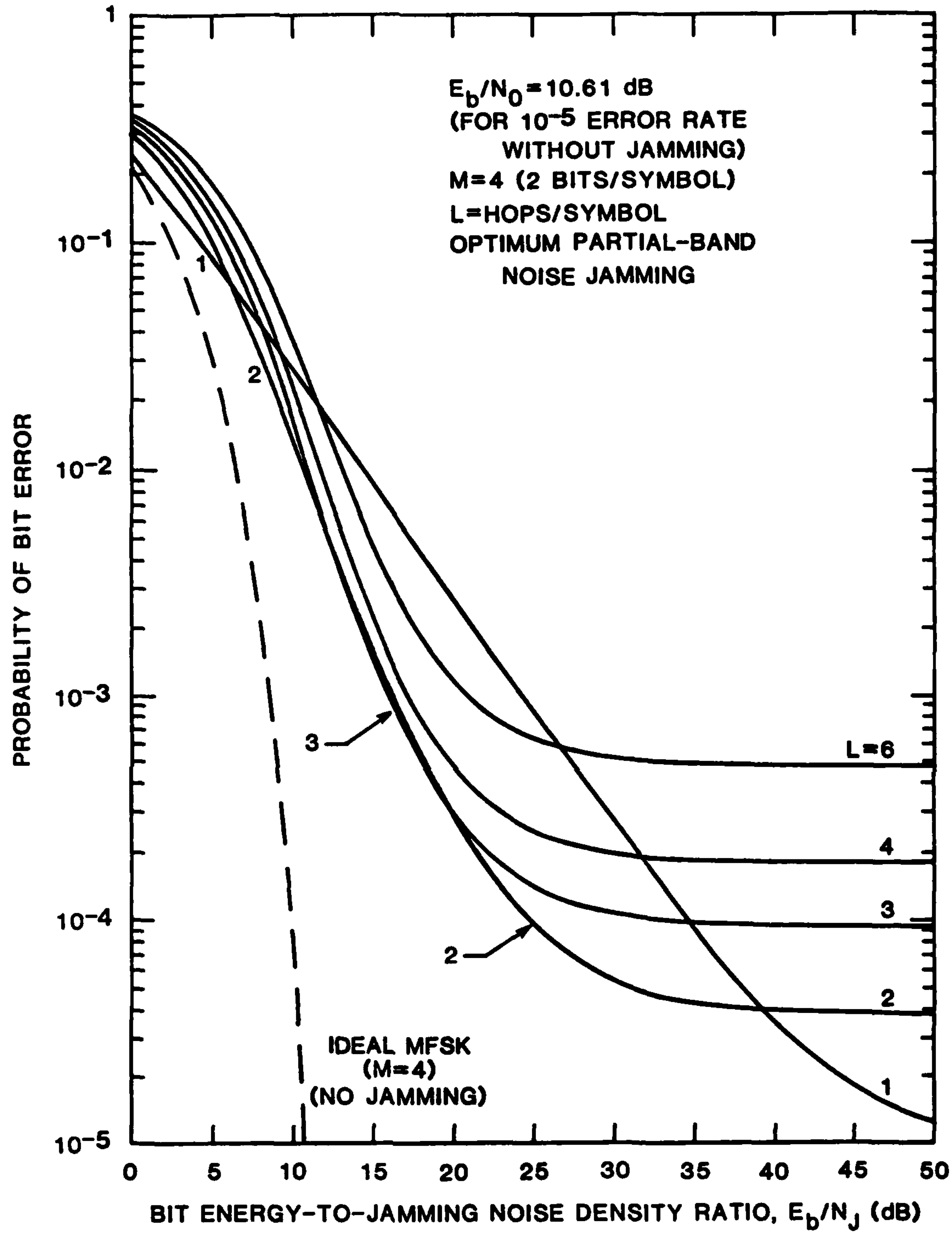


FIGURE 4-19 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF AGC FH/MFSK RECEIVER FOR M=4 WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER WHEN $E_b/N_0 = 10.61$ dB (FOR IDEAL MFSK (M=4) CURVE THE ABSCISSA READS E_b/N_0)

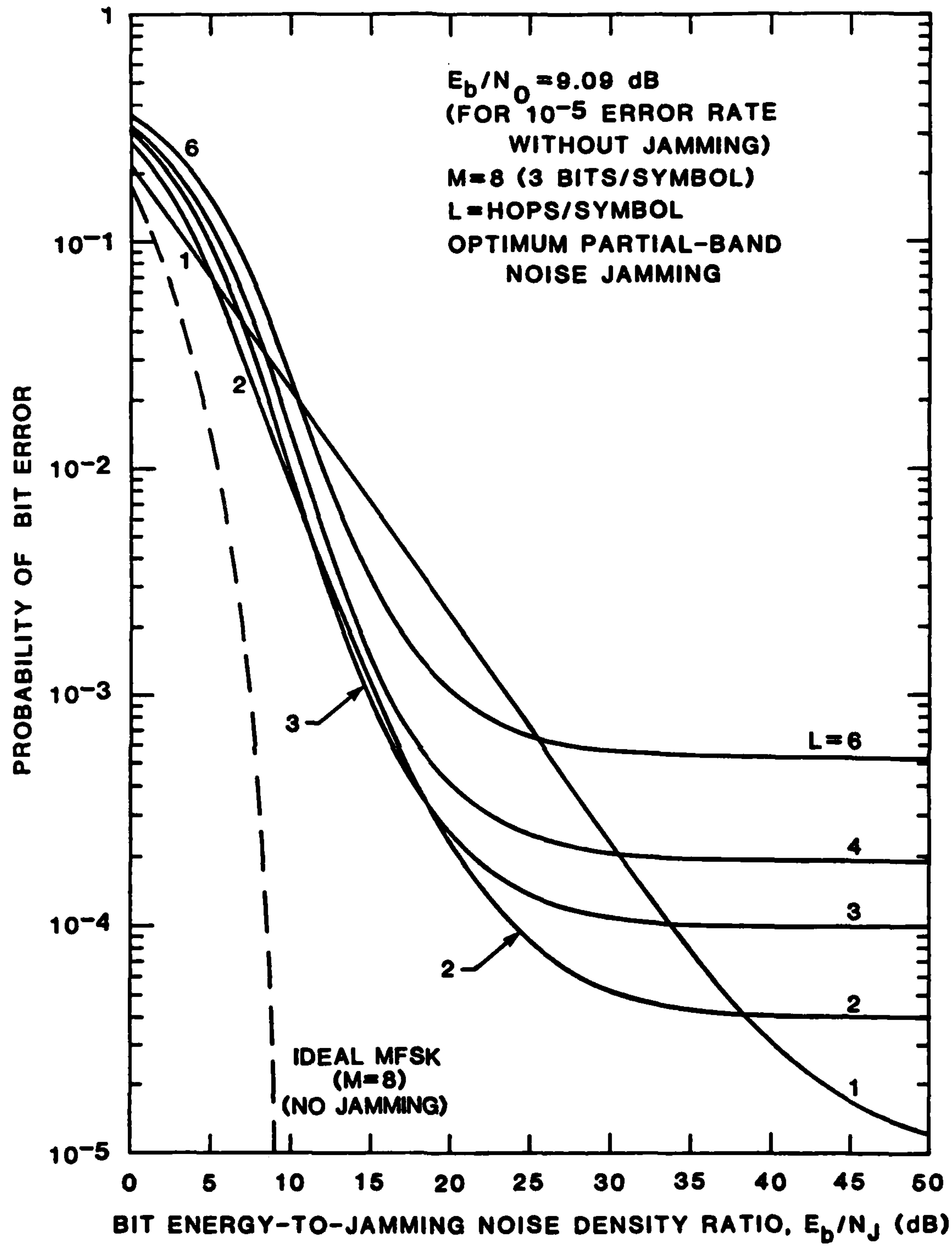


FIGURE 4-20 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF AGC FH/MFSK RECEIVER FOR M=8 WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER WHEN E_b/N_0=9.09 dB (FOR IDEAL MFSK (M=8) CURVE THE ABSCISSA READS E_b/N_0)

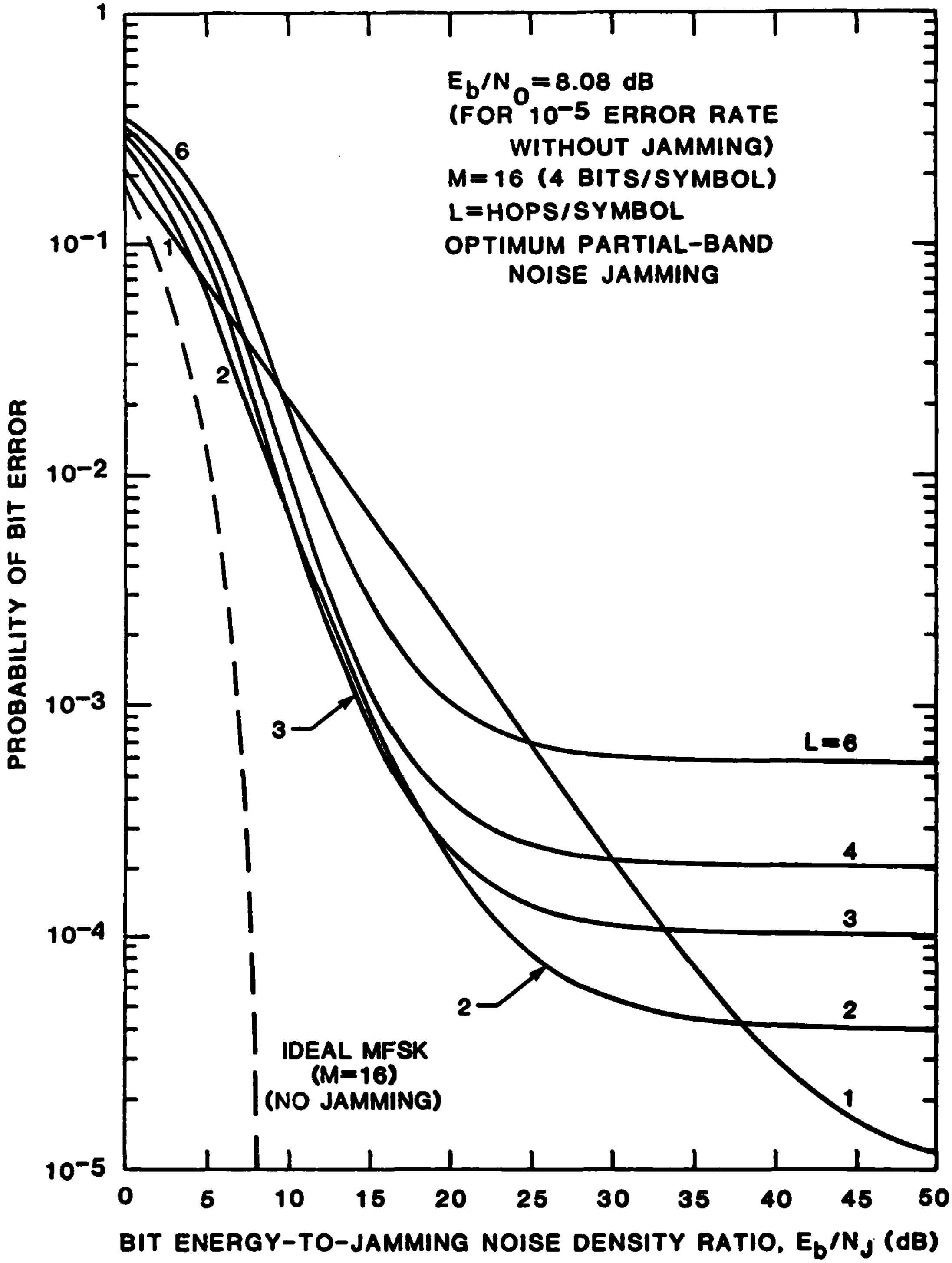


FIGURE 4-21 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF AGC FH/MFSK RECEIVER FOR M=16 WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER WHEN E_b/N_0 = 8.08 dB (FOR IDEAL MFSK (M=16) CURVE THE ABSCISSA READS E_b/N_0)

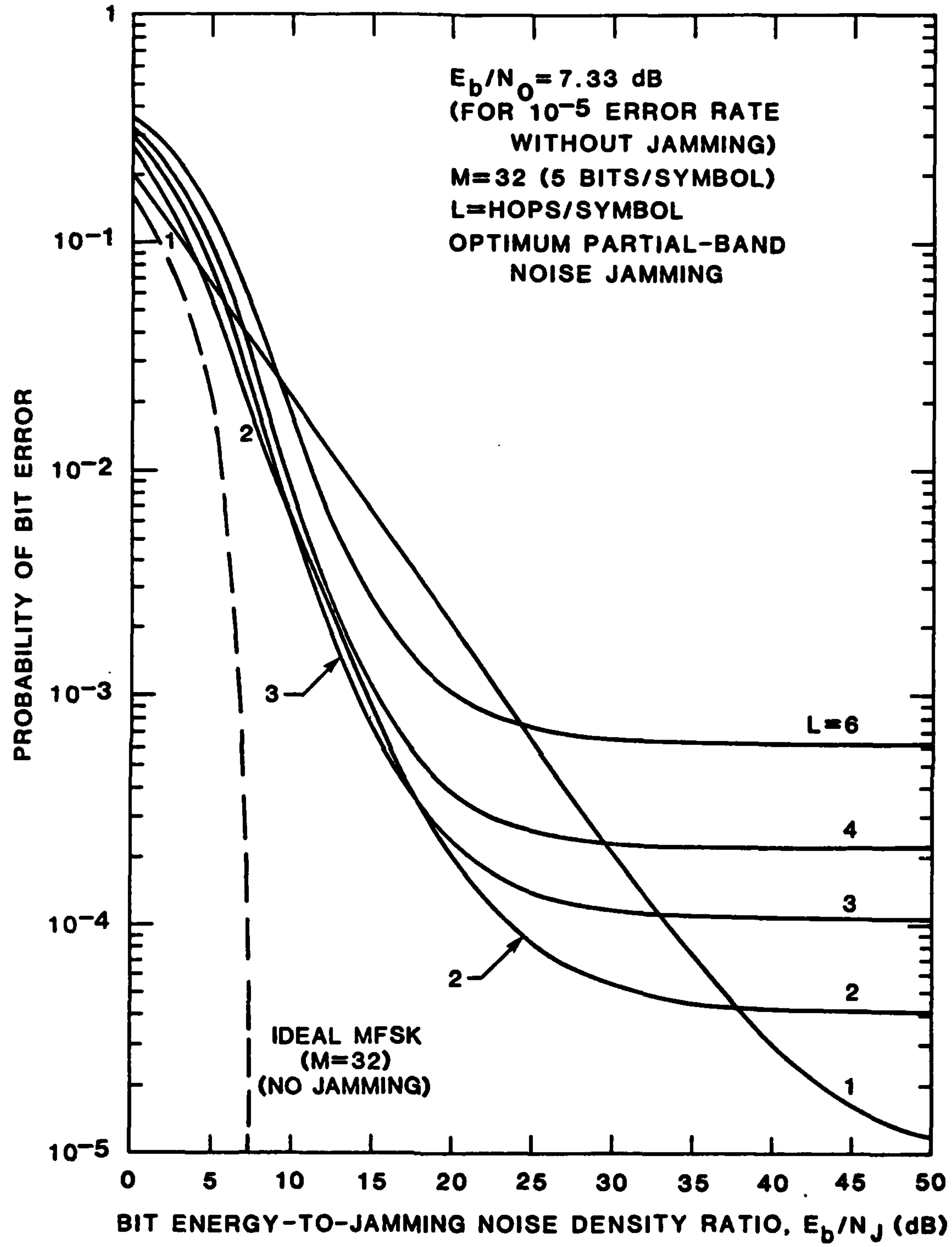


FIGURE 4-22 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF AGC FH/MFSK RECEIVER FOR M=32 WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER WHEN E_b/N_0=7.33 dB (FOR IDEAL MFSK (M=32) CURVE THE ABSCISSA READS E_b/N_0)

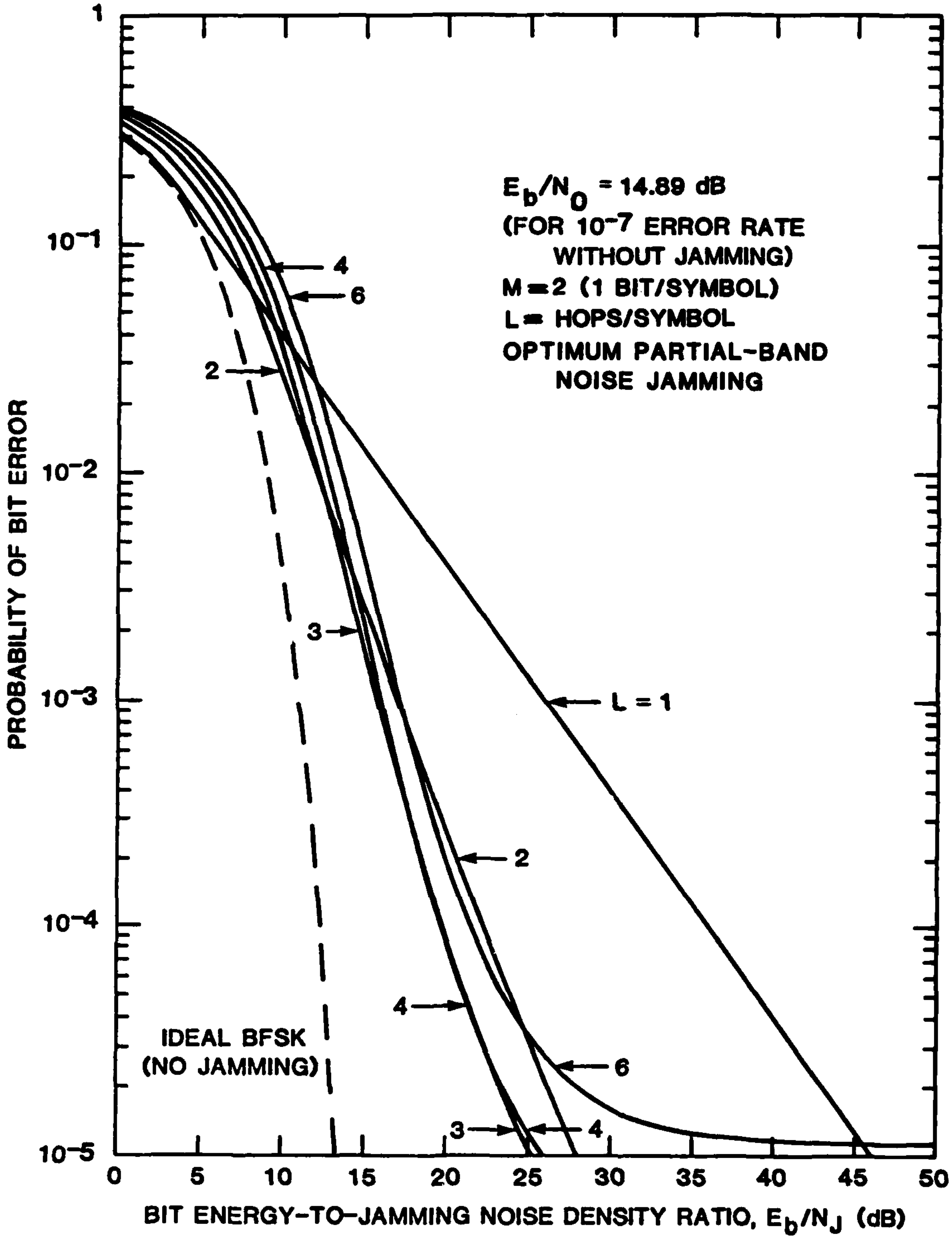


FIGURE 4-23 OPTIMUM JAMMING PERFORMANCE OF THE AGC RECEIVER FOR BFSK/FH WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER WHEN E_b/N_0 = 14.89 dB (FOR IDEAL BFSK CURVE THE ABSCISSA READS E_b/N_0)

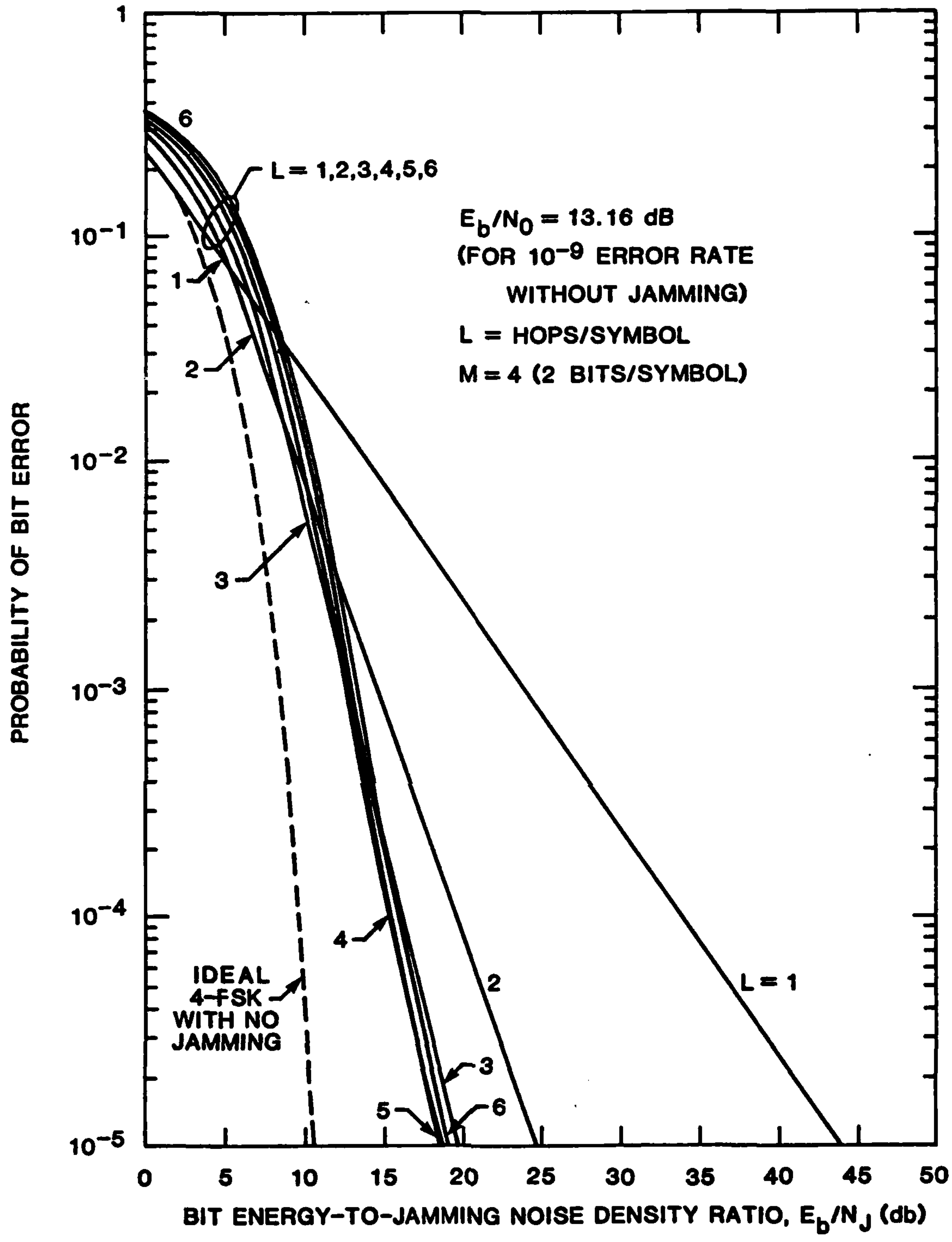


FIGURE 4-24 OPTIMUM JAMMING PERFORMANCE OF THE AGC FH/MFSK (M=4) RECEIVER WHEN E_b/N_0 = 13.61 dB WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER (FOR IDEAL MFSK (M=4) CURVE THE ABSCISSA READS E_b/N_0)

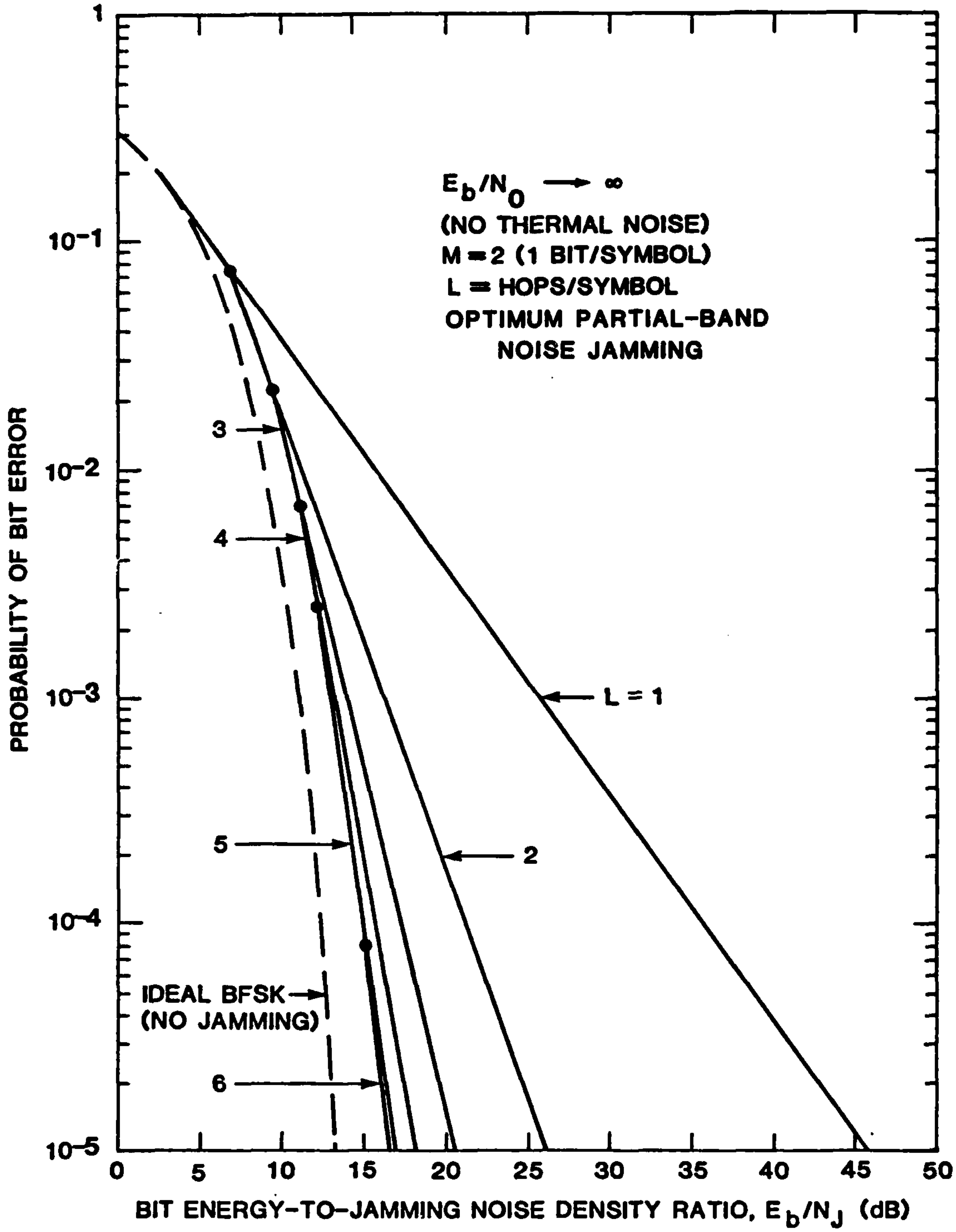


FIGURE 4-25 OPTIMUM JAMMING PERFORMANCE OF THE AGC RECEIVER FOR BFSK/FH WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER WHEN THERMAL NOISE IS ABSENT (FOR IDEAL BFSK CURVE THE ABSCISSA READS E_b/N_0)

4.4 PROBABILITY OF ERROR ANALYSIS FOR LINEAR-LAW AGC RECEIVER

It is well known that in the Gaussian noise channel detection of a sinusoidal signal based on accumulating square-law envelope samples yields a performance very similar to that based on accumulating linear-law envelope samples. Thus even when a linear-law envelope detector is employed the usual procedure is to analyze the problem as if square-law envelope detection were used, since the exact treatment of the linear-law case is not tractable. For FH/MFSK communications problems, it is not known in general whether a significant difference in performance between the two types of envelope detector is experienced, particularly when the system is jammed. Therefore, in this section we consider the performance of an AGC receiver for FH/MFSK in partial-band jamming using linear-law envelope detectors.

The system under consideration is shown in Figure 4-26. The symbol decison is made by selecting the largest of the decision statistics

$$z_i = \sum_{k=1}^{L} z_{ik} = \sum_{k=1}^{L} v_{ik}/\sigma_k, \quad i = 1, 2, \ldots, M, \tag{4-41}$$

where the v_{ik} are samples of the envelope detectors in the M channels at t_k, $k = 1, 2, \ldots, L$, corresponding to the L hops constituting the M-ary symbol and σ_k^2 is the noise power present on a given hop, assumed to be measured perfectly. As for the square-law envelope detector AGC receiver shown previously in Figure 4-1, the noise variances in the M dehopped channels are assumed to be equal on a given hop, with

$$\sigma_k^2 = \begin{cases} \sigma_N^2 = N_0 B & \text{with probability } 1-\gamma \\ \sigma_T^2 = (N_0 + N_J/\gamma)B & \text{with probability } \gamma, \end{cases} \tag{4-42}$$

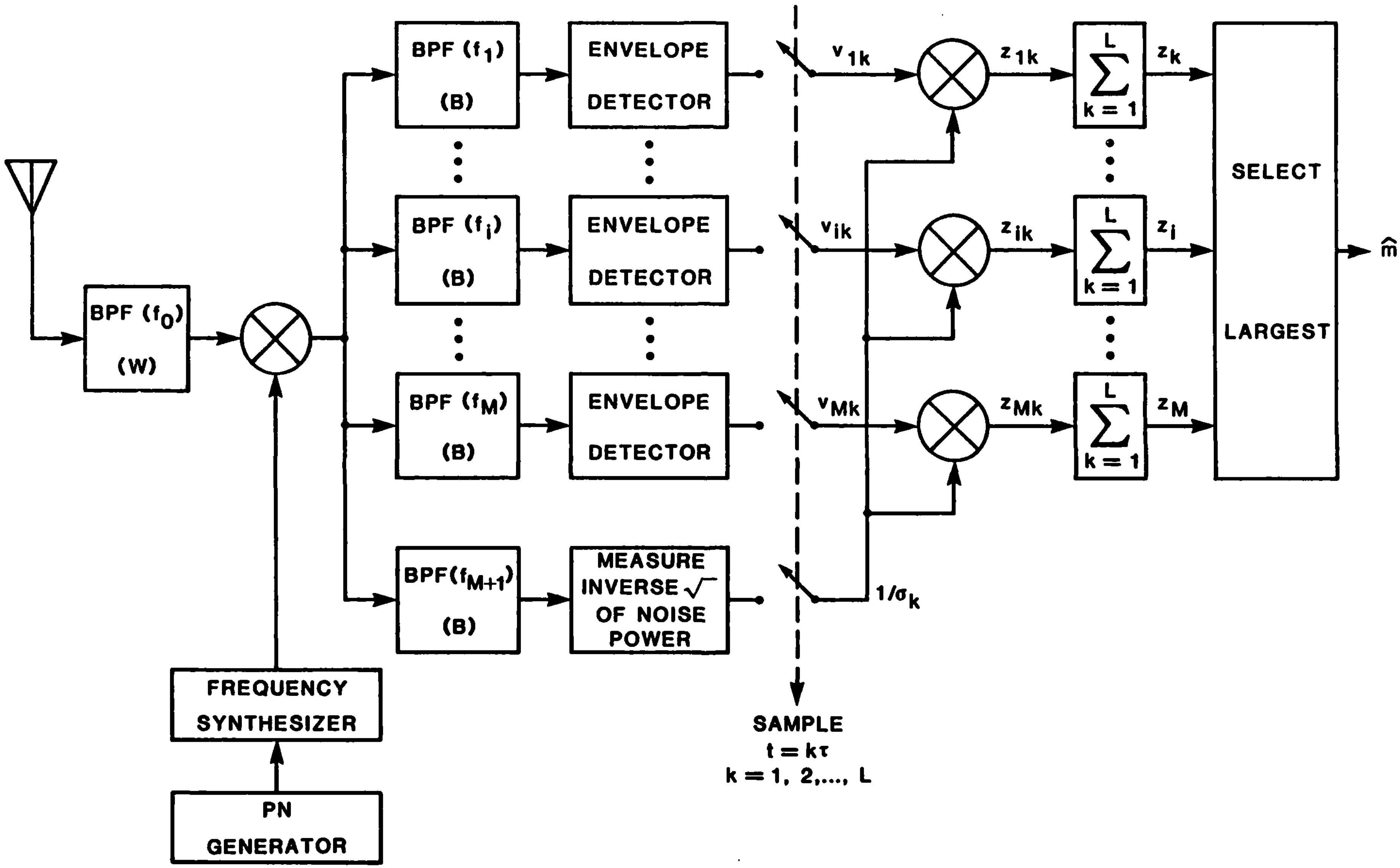


FIGURE 4-26 FH/MFSK RECEIVER WITH ADAPTIVE GAIN CONTROL EMPLOYING LINEAR LAW FOR USE IN PARTIAL-BAND JAMMING ENVIRONMENT

where γ is the fraction of the total hopped system bandwidth (W) which is jammed and N_J is the jammer's spectral density, averaged over W.

Intuitively, we might expect linear-law envelope detection to improve the anti-jam performance of the AGC receiver over that shown for the square-law system, since the weights (normalization) used on each hop are effectively

$$w_{ik} = \frac{1}{\sigma_k \sqrt{x_{ik}}}, \quad \left(\sqrt{x_{ik}} \equiv v_{ik}\right) \tag{4-43}$$

with respect to the square-law envelope samples x_{ik}, thus weighing jammed hops less than before (on the average, $\overline{x_{ik}} = 2(\sigma_k^2 + S)$).

As in Section 4.2, we express the bit error probability by

$$P_b(e) = \frac{M}{2(M-1)} \; P_s(e|m_1 \text{ transmitted})$$

$$= \frac{M}{2(M-1)} \sum_{\ell=0}^{L} \binom{L}{\ell} \gamma^{\ell}(1-\gamma)^{L-\ell} \; P_s(e|m_1, \ \ell \text{ hops jammed})$$

and calculate the conditional symbol error probabilities from

$$P_s(e|\ell) \equiv P_s(e|m_1, \ \ell \text{ hops jammed})$$

$$= 1 - \int_0^\infty d\alpha \; p_{z_1}(\alpha) \left[\int_0^\alpha d\beta \; p_{z_2}(\beta) \right]^{M-1}$$

$$= 1 - \int_0^\infty d\alpha \; p_{z_1}(\alpha) \left[F_{z_2}(\alpha) \right]^{M-1}, \tag{4-44}$$

in which $p_{z_1}(\alpha)$ and $p_{z_2}(\beta)$ are the probability density functions (pdf's) for signal plus noise and noise-only channels, respectively, and $F_{z_2}(\alpha)$ is the cumulative distribution function (cdf) for a noise-only channel,

$$F_{z_2}(\alpha) = \Pr(z_2 < \alpha). \tag{4-45}$$

4.4.1 Distribution of the Decision Statistics

From (4-41) and (4-11a), the decision statistics z_i are defined as

$$z_1 = \sum_{k=1}^{L} v_{1k}/\sigma_k = \sum_{k=1}^{L} \sqrt{x_{1k}/\sigma_k^2} \equiv \sum_{k=1}^{L} z_{1k}$$

$$= \sum_{k=1}^{L} \left\{ \left(\sqrt{\frac{2S}{\sigma_k^2}} \cos\theta_k + \nu_{c1k} \right)^2 + \left(\sqrt{\frac{2S}{\sigma_k^2}} \sin\theta_k + \nu_{s1k} \right)^2 \right\}^{\frac{1}{2}} \tag{4-46a}$$

and

$$z_i = \sum_{k=1}^{L} v_{ik}/\sigma_k^2 = \sum_{k=1}^{L} \sqrt{x_{ik}/\sigma_k^2} \equiv \sum_{k=1}^{L} z_{ik}$$

$$= \sum_{k=1}^{L} \sqrt{\nu_{cik}^2 + \nu_{sik}^2}, \quad i = 2, 3, \ldots, M. \tag{4-46b}$$

Since ν_{cik} and ν_{sik} are independent zero-mean unit-variance Gaussian random variables, conditionally the z_i for $i \geq 2$ are sums of L normalized Rayleigh random variables and z_1 is the sum of L normalized Rician random variables with SNR's $\rho_k = S/\sigma_k^2$; $k = 1, 2, \ldots, L$. Thus the pdf's of z_{1k} and z_{ik} are

$$p_{z_{1k}}(\alpha) = \alpha \exp\left\{-\rho_k - \alpha^2/2\right\} I_0\left(\alpha\sqrt{2\rho_k}\right) \tag{4-47a}$$

and

$$p_{z_{ik}}(\beta) = \beta e^{-\beta^2/2}, \; i = 2, 3, \ldots, M. \tag{4-47b}$$

For L=1 hop/symbol, the exact equations for the pdf and cdf of the decision variables z_i are known. For the case of L=2, exact analysis is still straightforward to compute, since the analytical expression for the pdf of the convolution of two Rayleigh densities can be easily obtained and the convolution of two Rician densities can be numerically evaluated without any difficulty. However, when $L > 2$ the analytical expression and numerical analysis of the exact pdf of the decision variables z_i become too complicated to obtain. To overcome this problem, we approximate the pdf and cdf of the decision variables z_i by asymptotic* expansions (Edgeworth series).

The Edgeworth series expresses the pdf of a standardized random variable X as the sum of derivatives of the Gaussian pdf, weighted by functions of the cumulants of X. For the FH/MFSK decision variables, then, we approximate the pdf's by the asymptotic expansions

$$p_{x_i}(\alpha) = \frac{1}{\sigma_{z_i}} p_X\left(\frac{\alpha - \overline{z}_i}{\sigma_{z_i}}\right)$$

$$\sim \frac{1}{\sigma_{z_1}} \sum_n c_{ni} Z^{(n)}\left(\frac{\alpha - \overline{z}_i}{\sigma_{z_i}}\right); \qquad i = 1, 2, \ldots, M; \tag{4-48}$$

where the $Z^{(n)}(\cdot)$ are derivatives of the Gaussian pdf. Similarly the cdf of the decision statistics can be expressed as

*Asymptotic in L; hence the accuracy of the approximation increases as L increases.

$$F_{z_i}(\alpha) = F_x\left(\frac{\alpha - \bar{z}_i}{\sigma_{z_i}}\right)$$

$$\sim \sum_n c_{ni}\, Z^{(n-1)} \left(\frac{\alpha - \bar{z}_i}{\sigma_{z_i}}\right). \qquad (4\text{-}49)$$

Equations (4-48) and (4-49) are now expressed in terms of a random variable x which can be evaluated by the expansion technique explained in Appendix 4B and applied to the problem at hand in Appendix 4C. The bit error probability then can be obtained by substituting (4-48) and (4-49) into (4-44), to give the expression

$$P_b(e) = \frac{M/2}{M-1} \sum_{\ell=0}^{L} \binom{L}{\ell} \gamma^{\ell} (1-\gamma)^{L-\ell}$$

$$\times \left\{ 1 - \int_0^{\infty} d\alpha \, \frac{1}{\sigma_{z_1}} \, p_x\left(\frac{\alpha - \bar{z}_1}{\sigma_{z_1}}\right) \left[F_x\left(\frac{\alpha - \bar{z}_2}{\sigma_{z_2}}\right)\right]^{M-1} \right\}. \qquad (4\text{-}50)$$

The results for L=4 and L=6 for given values of M (2, 4, and 8) are obtained by evaluating (4-50). For the case of L=1 and L=2, the bit error probabilities are obtained by the exact analysis given in Appendix 4D. For a given number of jammed hops (ℓ), the moments and cumulants of z_1 are computed using $\rho_k = S/\sigma_T^2 \equiv \rho_T$ for ℓ of the hops and $\rho_k = S/\sigma_N^2 \equiv \rho_N$ for L-ℓ of the hops.

4.4.2 Numerical Results for Linear-Law AGC Receiver

The worst-case or maximum probability of error is obtained by using the computer programs of Appendices 4F, 4G, and 4H to compute $P_b(e)$ while varying the fraction γ. A sample plot is shown in Figure 4-27 for M=4, L=1. For comparison purposes, the bit error probabilities for the two AGC receivers (linear and quadratic detectors) are plotted as a function of

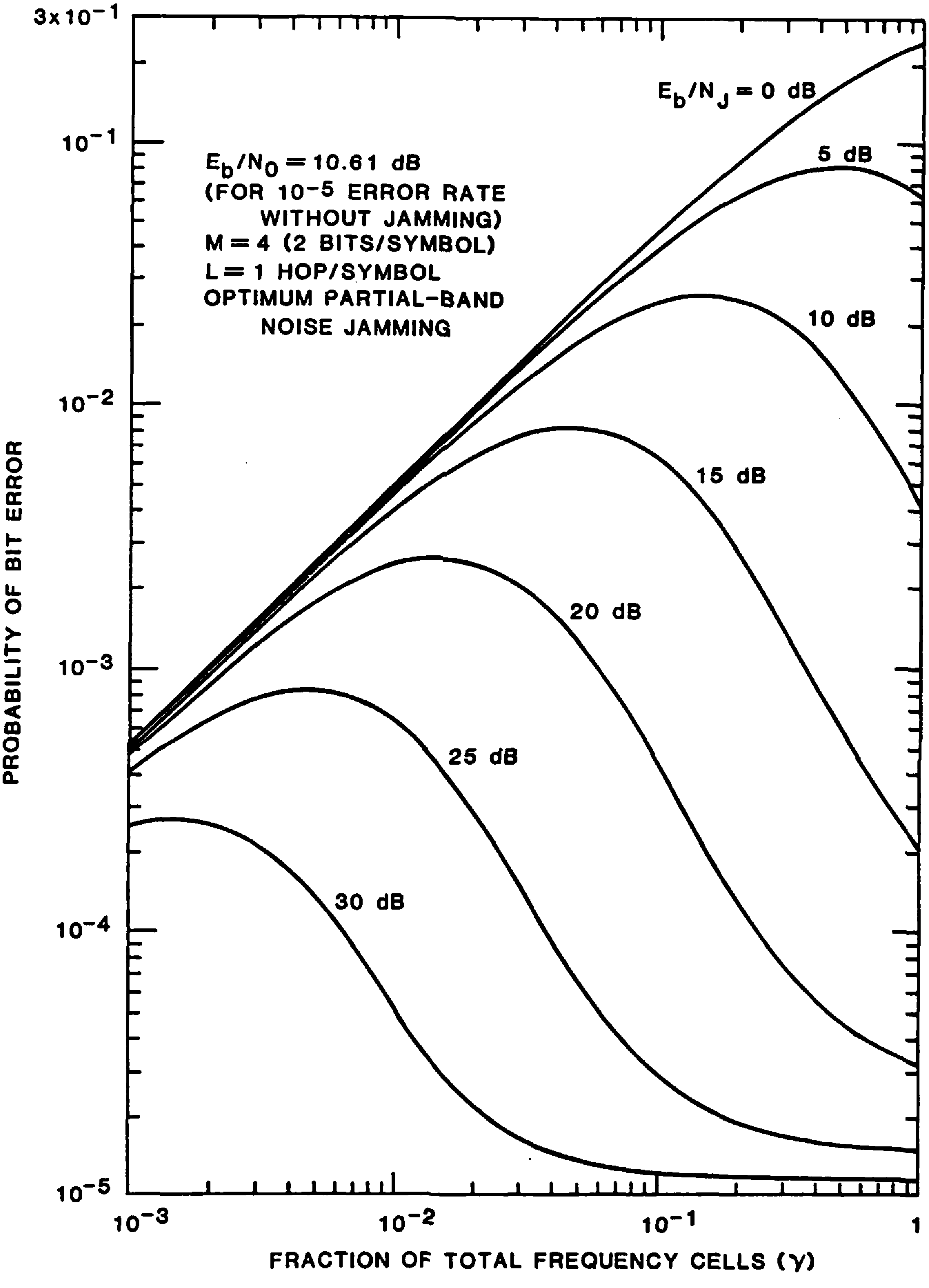


FIGURE 4-27 PROBABILITY OF BIT ERROR VS. FRACTION OF TOTAL FREQUENCY CELLS JAMMED FOR FH/MFSK (M = 4) AGC LINEAR-LAW RECEIVER WITH L = 1 HOP/SYMBOL WHEN $E_b/N_0 = 10.61$ dB

γ for M=4, L=6, in Figure 4-28. It is seen that the optimum γ decreases with increasing E_b/N_J and that when L is increased, the optimum γ is brought closer to 1. From the comparison of the two AGC receivers in Figure 4-28, it is seen that the performance difference is very much a function of E_b/N_J and fraction γ. For large values of E_b/N_J, the performance difference is rather insensitive to the fraction γ. In this region, for the same value of γ the linear-law detector always performs better than the square-law detector. However, for very low E_b/N_J, there is a cross-over point in γ, above which the square-law detector performs better and below which the linear-law performs better. The cross-over disappears when E_b/N_J is 10 dB or greater. However, when L is decreased, the cross-over disappears for higher values of E_b/N_J, at which point the linear-law detector will always perform better than the square-law detector.

Figures 4-29 through 4-31 show the worst-case bit error probability as a function of E_b/N_J for M=2, 4, and 8 and E_b/N_0's corresponding to 10^{-5} error rate, each for different values of L (L=1, 2, 4, and 6). The noncoherent combining loss is clearly illustrated for large values of E_b/N_J whereas for the range of E_b/N_J between, say, 5 dB and 40 dB, the diversity improvement is obvious by the comparison with the L=1 curves. This is due to the antijam capabilities of the AGC receiver. The square-law detector performance of the AGC receiver is also plotted for comparison. It is seen that the linear-law detector reduces the non-coherent combining loss for large values of E_b/N_J. The behavior of both receivers seems to be quite similar.

Figures 4-32 through 4-35 show the worst-case bit error probability as a function of E_b/N_J for L=1, 2, 4, and 6 each for different values of M (M=2, 4, and 8) and the corresponding E_b/N_0 for a 10^{-5} error rate. For L=1 the results are identical with the results given in Section 4.3. The NCL

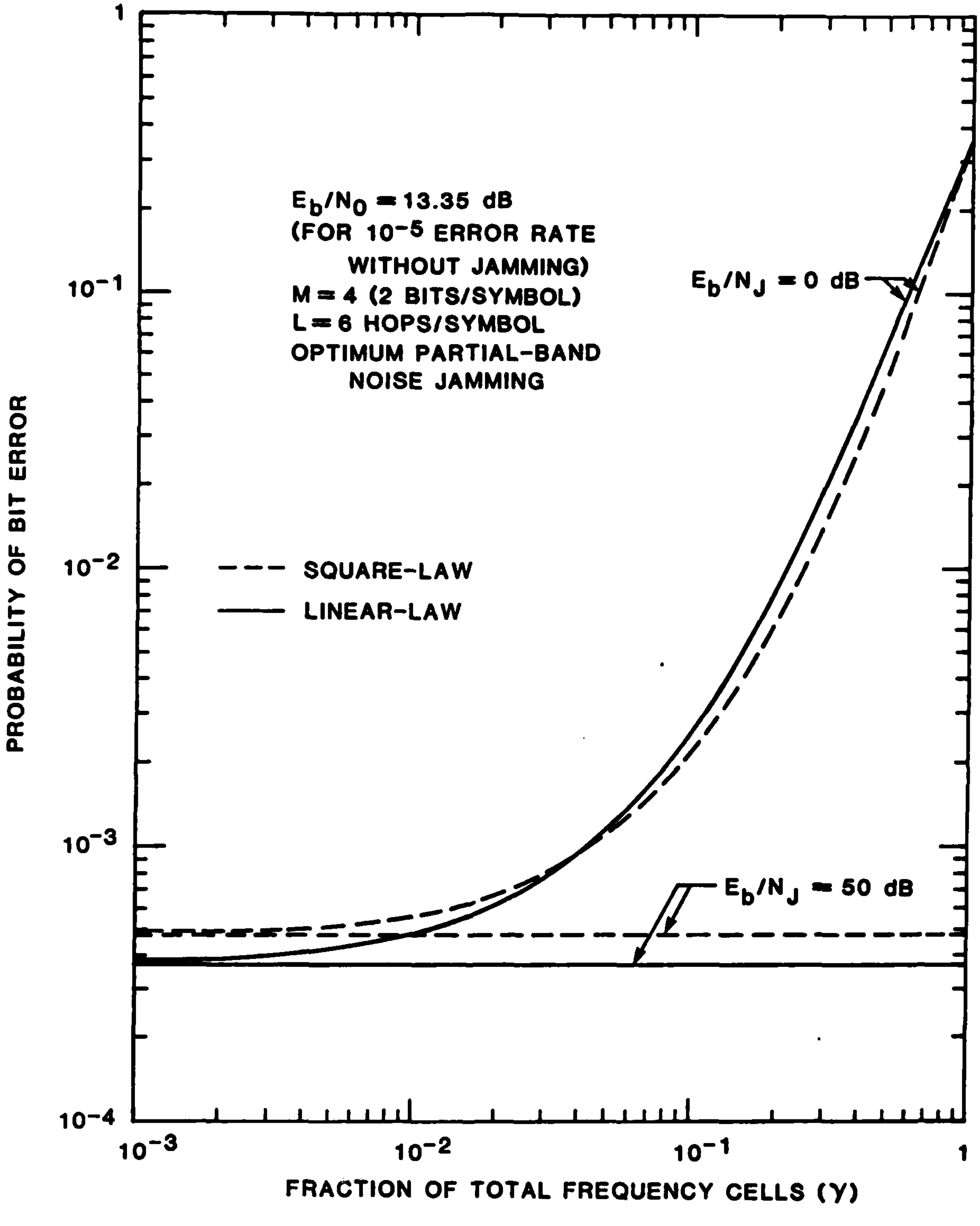


FIGURE 4-28 PROBABILITY OF BIT ERROR VS. FRACTION OF TOTAL FREQUENCY CELLS JAMMED FOR FH/MFSK (M = 4) AGC LINEAR-LAW RECEIVER WITH L = 6 HOPS/SYMBOL WHEN $E_b/N_0 = 10.61$ dB

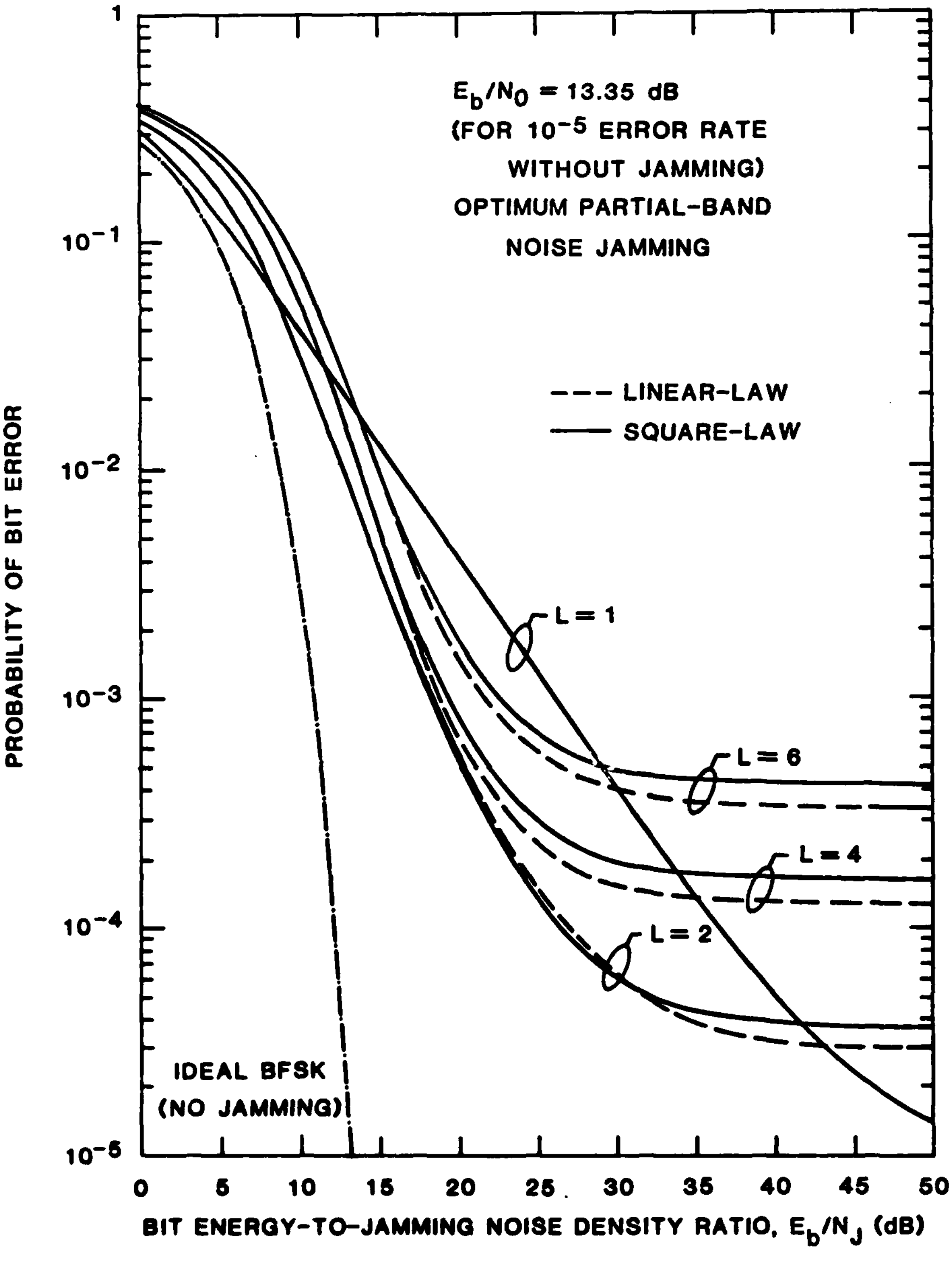


FIGURE 4-29 COMPARISON OF LINEAR-LAW AND SQUARE-LAW UNDER OPTIMUM JAMMING PERFORMANCE OF THE AGC RECEIVER FOR BFSK/FH WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER WHEN E_b/N_0 = 13.35 dB (FOR IDEAL BFSK CURVE THE ABSCISSA READS E_b/N_0)

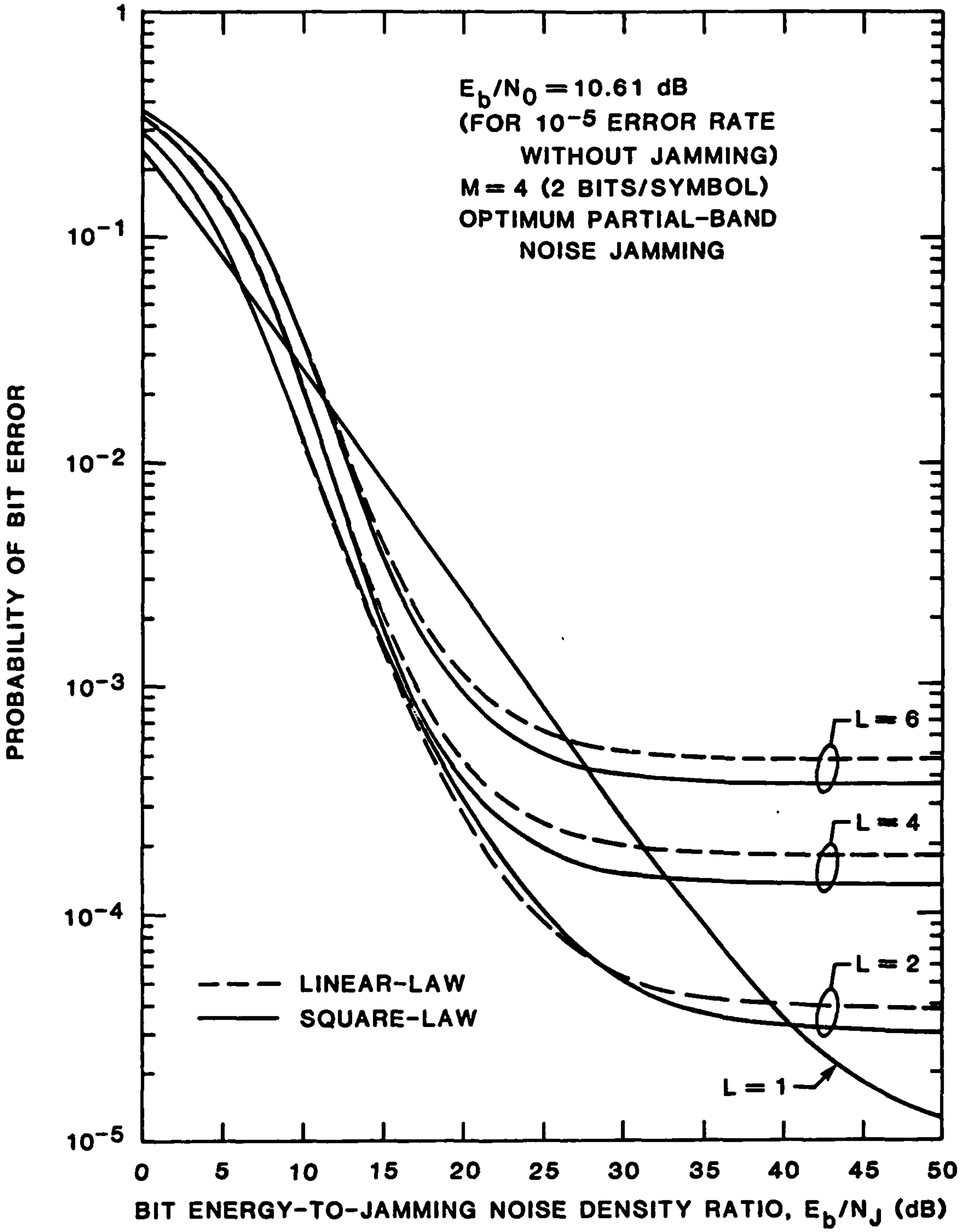


FIGURE 4-30 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE COMPARISONS OF FH/MFSK (M = 4) AGC LINEAR-LAW AND SQUARE-LAW RECEIVERS WHEN $E_b/N_0 = 10.61$ dB WITH THE NUMBER OF HOPS/SYMBOL (L) VARIED

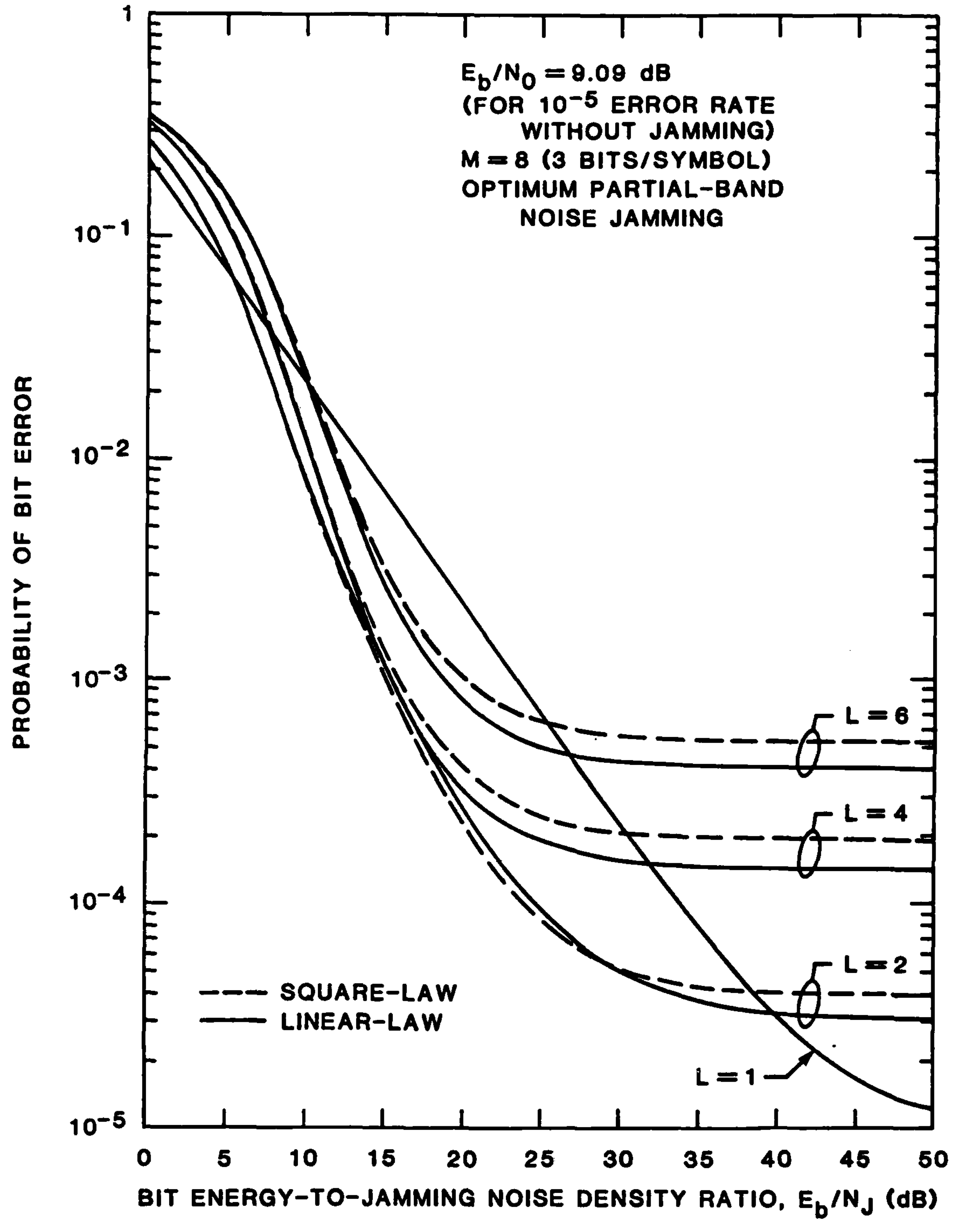


FIGURE 4-31 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE COMPARISONS OF FH/MFSK (M = 8) AGC LINEAR-LAW AND SQUARE-LAW RECEIVERS WHEN $E_b/N_0 = 9.09$ dB WITH THE NUMBER OF HOPS/SYMBOL (L) VARIED

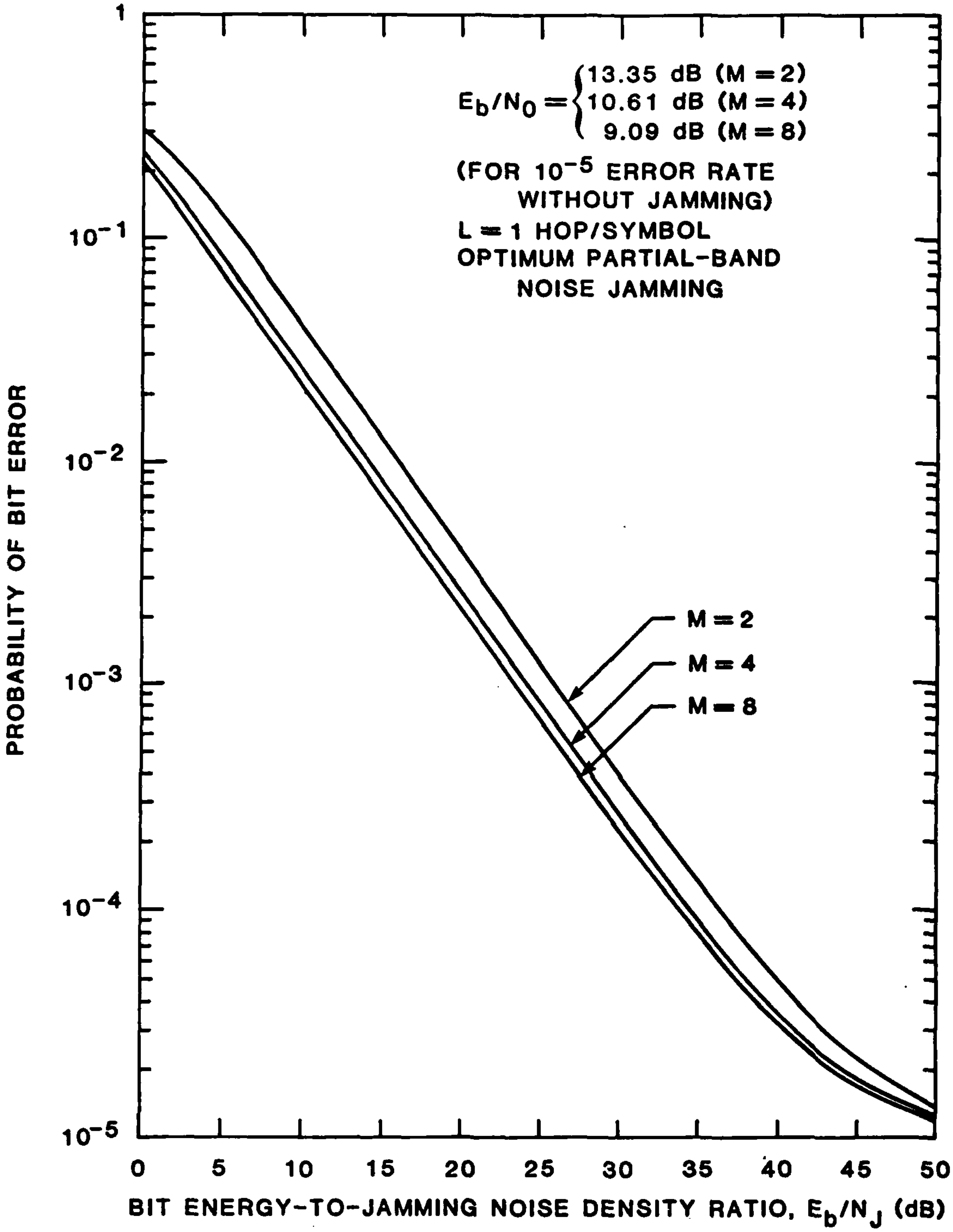


FIGURE 4-32 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE COMPARISONS OF FH/MFSK AGC LINEAR-LAW RECEIVER FOR L = 1 HOP/SYMBOL WITH THE ALPHABET SIZE (M) VARIED

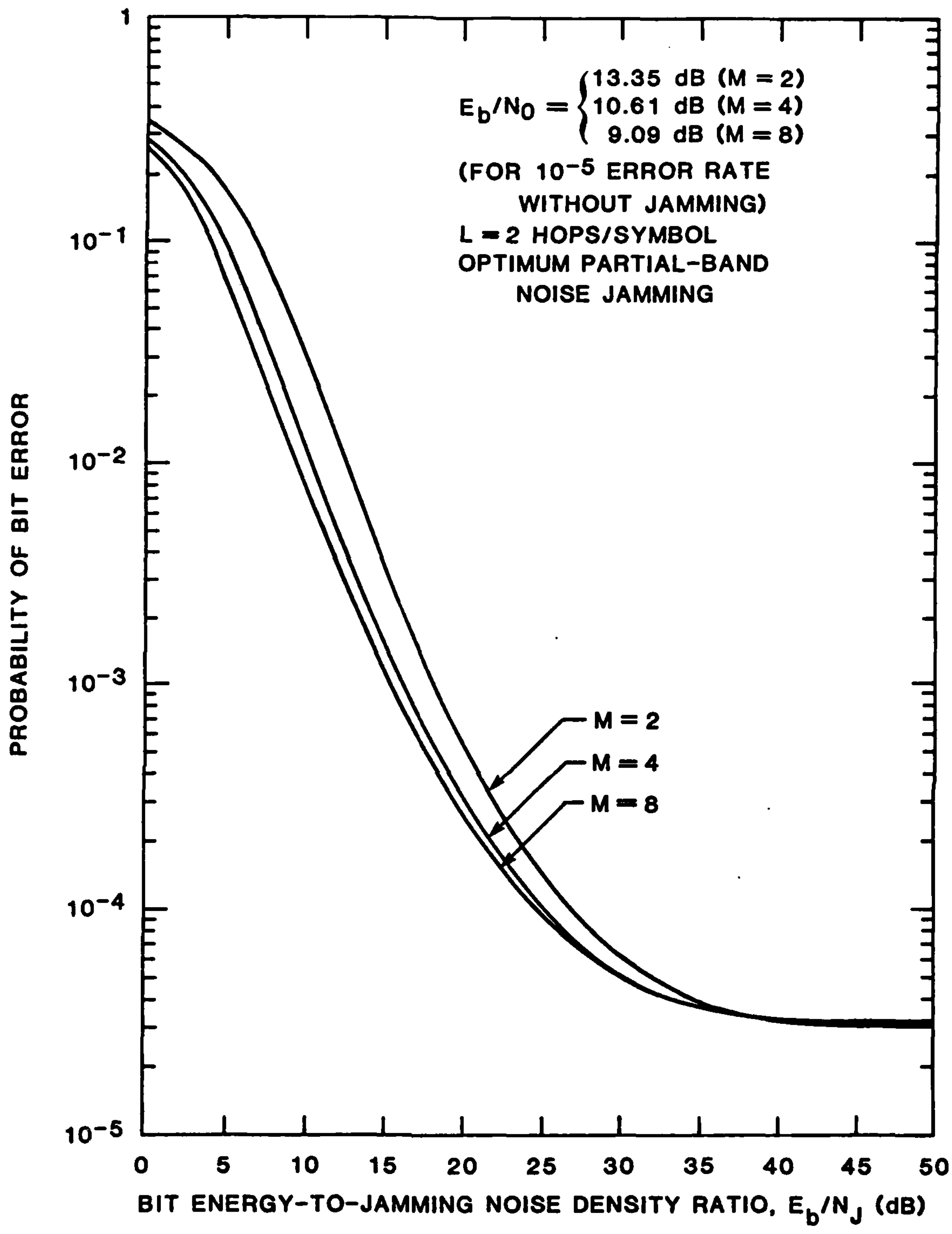


FIGURE 4-33 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE COMPARISONS OF FH/MFSK AGC LINEAR-LAW RECEIVER FOR L = 2 HOPS/SYMBOL WITH THE ALPHABET SIZE (M) VARIED

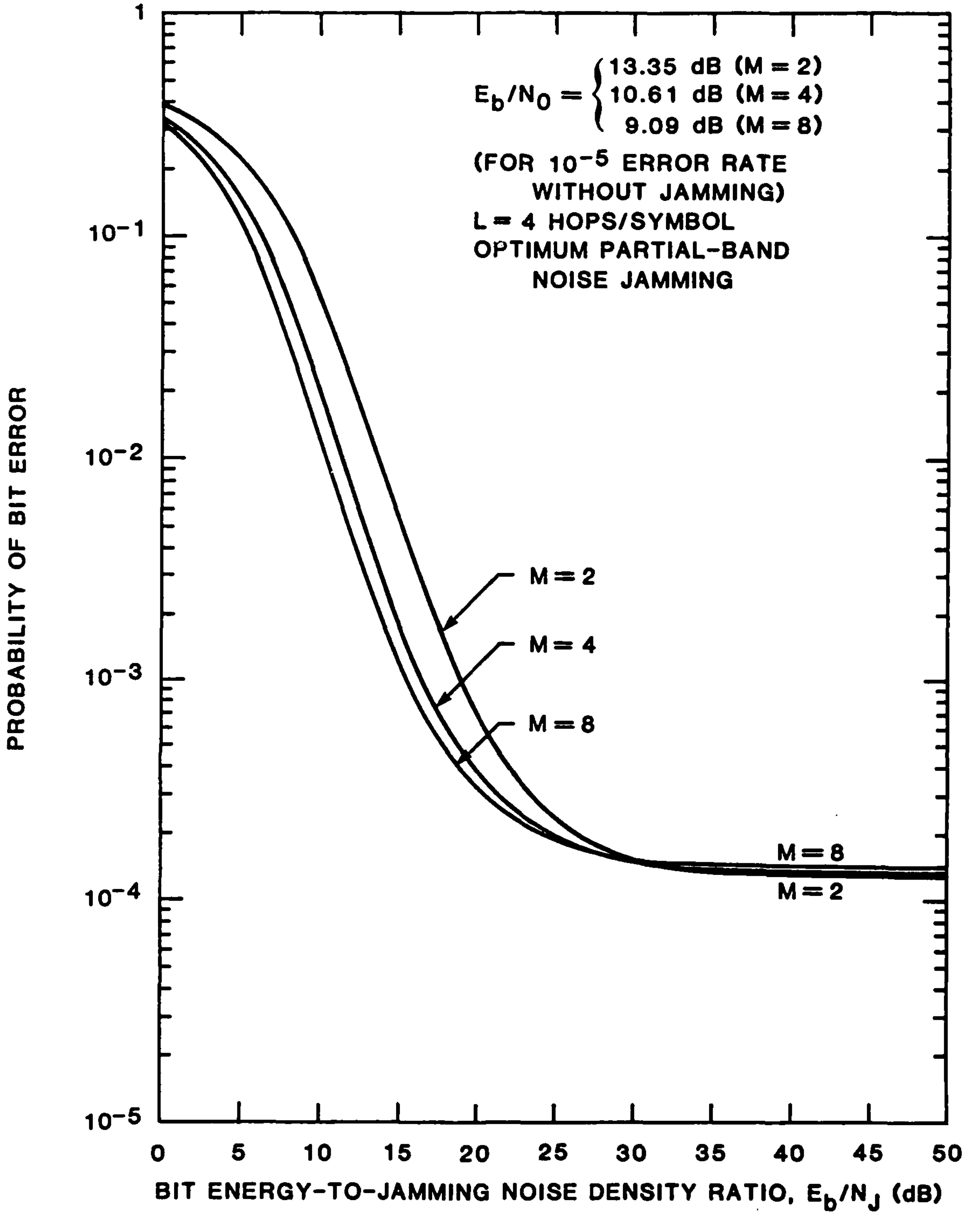


FIGURE 4-34 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE COMPARISONS OF FH/MFSK AGC LINEAR-LAW RECEIVER FOR L = 4 HOPS/SYMBOL WITH THE ALPHABET SIZE (M) VARIED

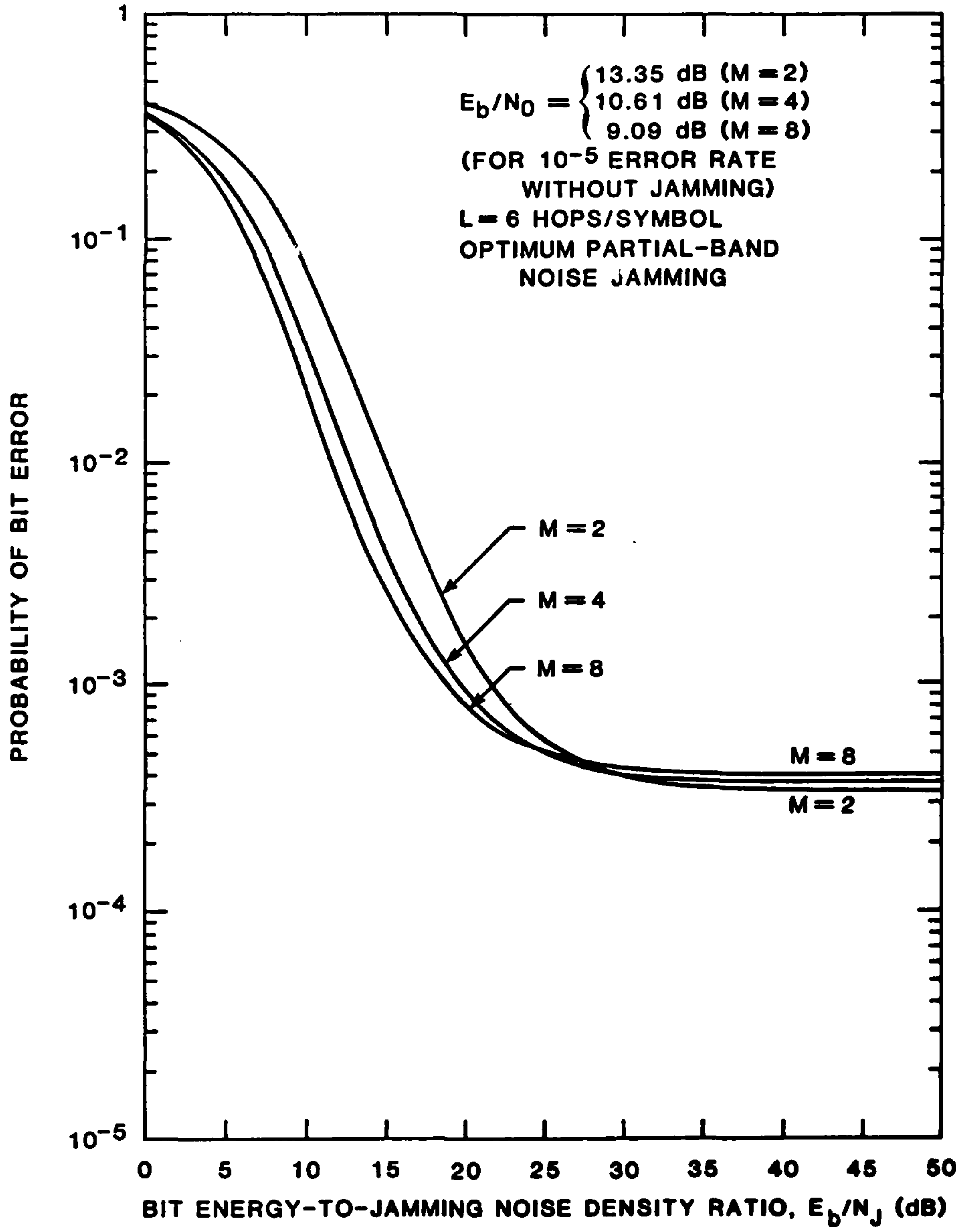


FIGURE 4-35 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE COMPARISONS OF FH/MFSK AGC LINEAR-LAW RECEIVER FOR L = 6 HOPS/SYMBOL WITH THE ALPHABET SIZE (M) VARIED

is not defined for L=1 hop/symbol. For large values of E_b/N_J, the $P_b(e)$ will read 10^{-5} for any given M under the above condition. However, for L=2 or greater, there is a cross-over point in E_b/N_J above which a lower value of M would perform better and below which a larger value of M would perform better. It is seen that for increasing values of L, the cross-over moves to the left. It is also seen that when L is increased, the difference in NCL becomes more distinctive.

Finally the wideband performance for a typical AGC linear-law receiver for M=2 and L=2 is shown for purpose of comparison with the performance of an AGC square-law receiver in Figure 4-36. This is taken as a typical example since, for L=4 or greater, the performances under optimum partial-band jamming are very much equal to the performances under wideband jamming. It is seen that for E_b/N_J greater than 5 dB the linear-law receiver performs better than the square-law receiver.

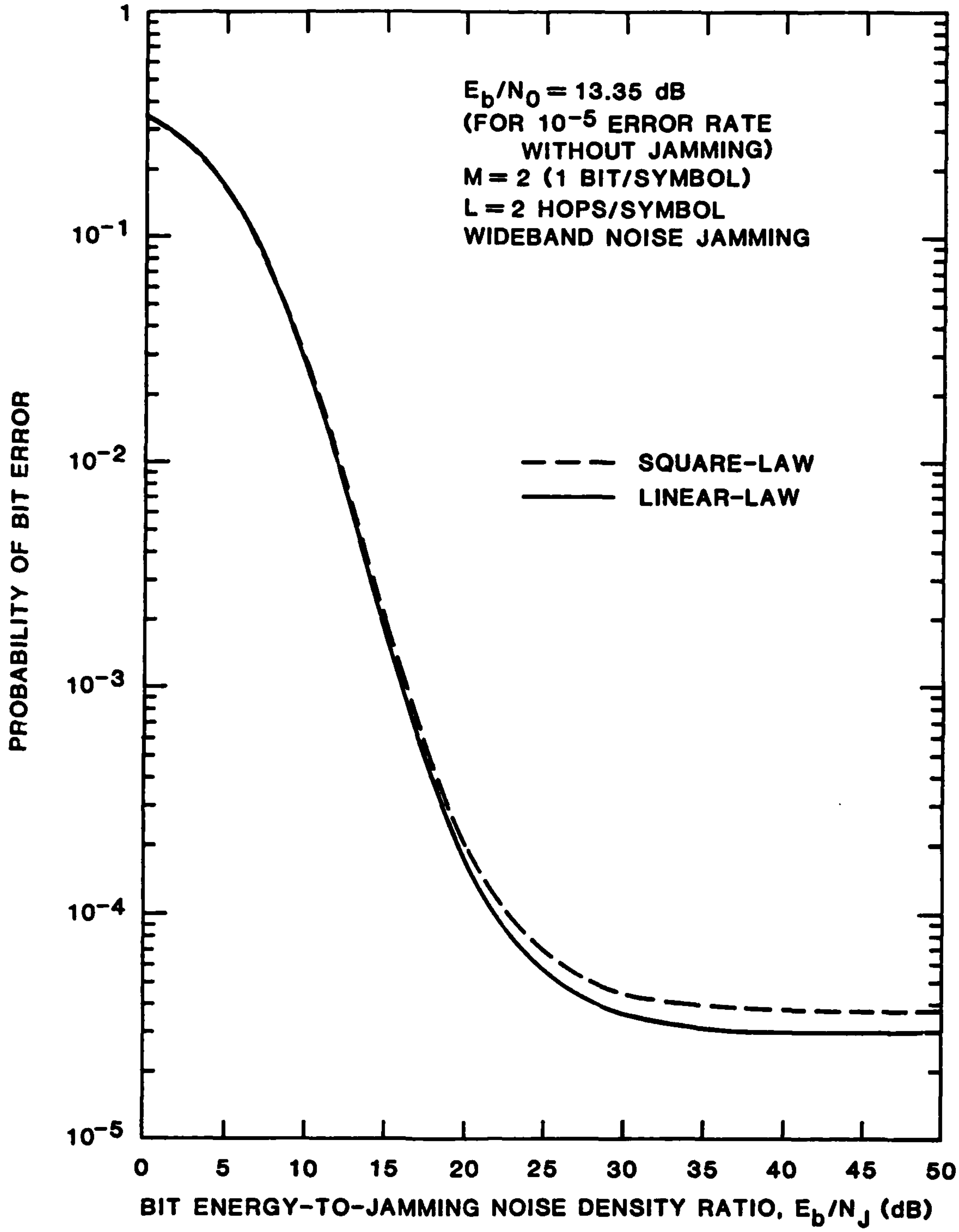


FIGURE 4-36 WIDEBAND NOISE JAMMING PERFORMANCE COMPARISONS OF FH/MFSK (M = 2) AGC LINEAR-LAW AND SQUARE-LAW RECEIVERS FOR L = 2 HOPS/SYMBOL WHEN $E_b/N_0 = 13.35$ dB

5.0 PARTIAL-BAND NOISE JAMMING PERFORMANCE OF AN L-HOPS PER SYMBOL FH/MFSK RECEIVER EMPLOYING SELF-NORMALIZATION

In Sections 3.0 and 4.0 we have shown by the results of exact error probability analyses that two types of square-law non-linear combining (AGC and clipper) receivers for FH/MFSK are effective in mitigating optimum partial-band noise jamming for $L > 1$. The AGC receiver analysis assumed that a perfect estimate of the noise power on each hop was available for the purpose of normalizing the detector outputs. Due to this idealized normalization of the detector outputs, the performance of the receiver is useful as a lower bound on what may be realized in practice.

We now consider the error performance of another, more practical, class of square-law combining receiver for FH/MFSK signals in the partial-band noise jamming environment. The normalizations of this receiver are provided by the samples taken at the output of square-law envelope detectors during a single hop period. We shall call this the self-normalizing receiver, owing to its self-sufficiency in normalizing the detector outputs. One of the advantages of this receiver is that no extra channel is needed to normalize the detector outputs and, therefore, it is easier to implement. Since the self-normalizing receiver resembles the AGC receiver in that both receivers weight the sample outputs, we expect that the weights W_k generated by the self-normalizing receiver will provide an anti-jam (AJ) capability against optimum partial-band jamming for $L > 1$.

In the following, we give a brief description of the system model, then proceed with the analysis of the error probability in general for the M-ary case. In subsection 5.3, we consider a special case of M=2. We consider both a receiver without a quantizer and one with a finite N-level quantizer for linear and square-law detectors in anticipation of a digital implementation.

The effect of the quantization level is discussed and finally, in subsection 5.4, we present numerical results for the receiver performance.

5.1 SYSTEM MODEL

The L-hops/symbol square-law combining self-normalizing receiver for FH/MFSK signals is modelled as shown in Figure 5-1. On a given hop the MFSK signal $s(t)$ is assumed to be one of M tones:

$$s(t) = \sqrt{2S}\cos(2\pi f_i t + \theta_k), \quad (k-1)\tau < t \leq k\tau,$$

$$k = 1, 2, \ldots, L, \quad i = 1, 2, \ldots, M, \tag{5-1}$$

where S is the received (average) signal power; f_i, $i = 1, 2, \ldots, M$, are the channel center frequencies; and θ_k, $k = 1, 2, \ldots, L$, are independent phases uniformly distributed on $[0, 2\pi)$.

We also assume that both thermal and jamming noise in any selected cell are stationary bandlimited white Gaussian noise. Using the Rician decomposition, we can write

$$n_i(t) = n_{ci}(t)\cos 2\pi f_i t + n_{si}(t)\sin 2\pi f_i t;$$

$$j_i(t) = j_{ci}(t)\cos 2\pi f_i t + j_{si}(t)\sin 2\pi f_i t,$$

$$i = 1, 2, \ldots, M; \tag{5-2}$$

where $n_{ci}(t)$, $n_{si}(t)$, $j_{ci}(t)$, and $j_{si}(t)$ at a given time are statistically independent Gaussian random variables with variances (or average power) given by

$$E\left[n_i^2(t)\right] = E\left[n_{ci}^2(t)\right] = E\left[n_{si}^2(t)\right] = \sigma_N^2,$$

$$E\left[j_i^2(t)\right] = E\left[j_{ci}^2(t)\right] = E\left[j_{si}^2(t)\right] = \sigma_J^2. \tag{5-3}$$

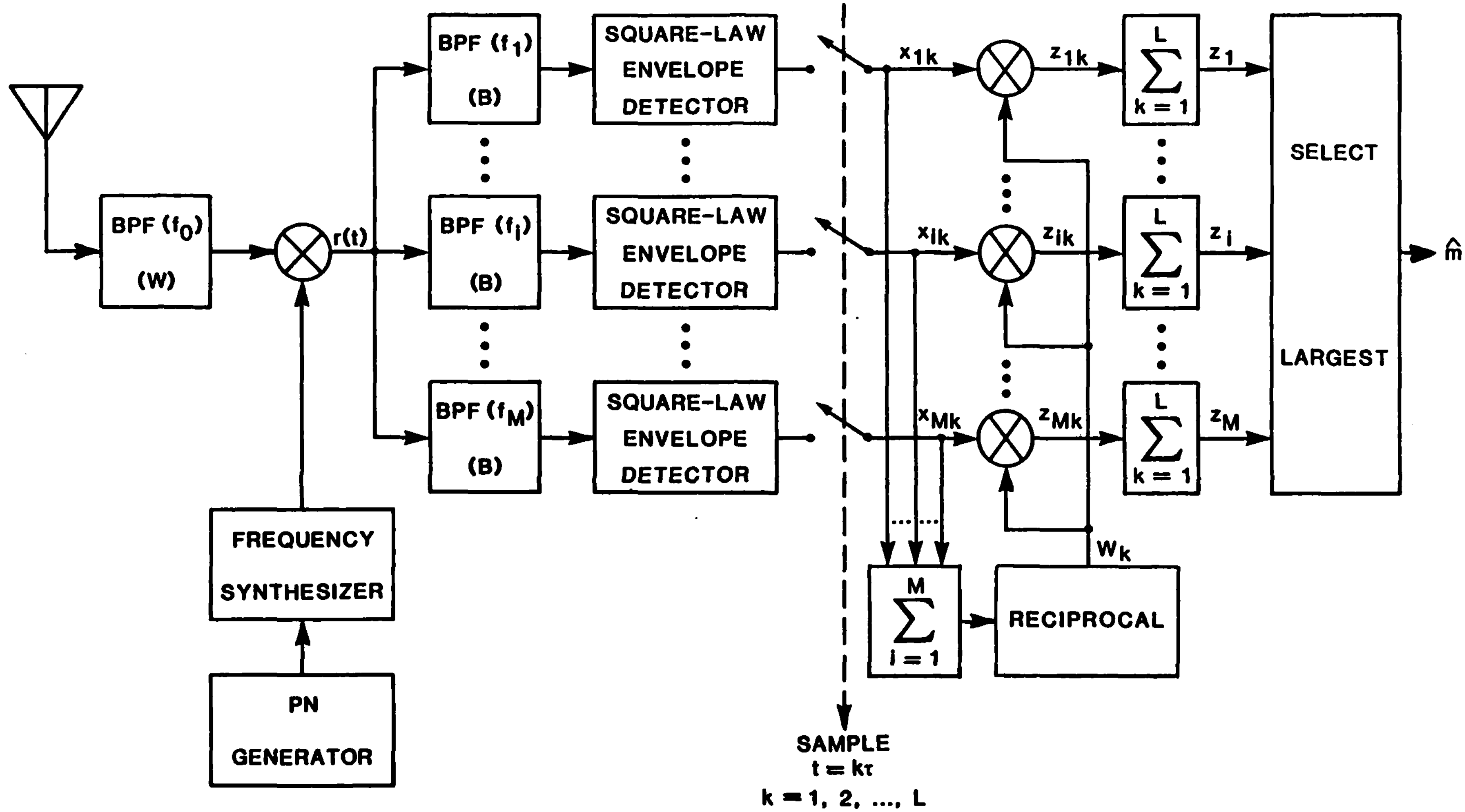


FIGURE 5-1 L HOPS/SYMBOL FH/MFSK SQUARE-LAW COMBINING SELF-NORMALIZING RECEIVER

Thermal noise is assumed to be present at all times, whereas jamming noise may or may not be present on a given hop. The jamming model used here assumes that each of the hops has the same probability γ of being jammed and the same probability $(1-\gamma)$ of not being jammed. Since $n_i(t)$ and $j_i(t)$ are additive noises, the resultant noise power σ^2 at the inputs to the envelope detectors may be written as

$$\sigma^2 = \begin{cases} \sigma_N^2, & \text{with probability } (1-\gamma) \\ \sigma_T^2 = \sigma_N^2 + \sigma_J^2, & \text{with probability } \gamma. \end{cases} \tag{5-4}$$

At the receiver front-end, the dehopped signal $r(t)$ is fed to M bandpass filters with center frequencies f_i, $i = 1, 2, \ldots, M$. After filtering, the receiver employs square-law envelope detectors whose outputs are sampled once every hop period τ to produce x_{ik} where, in channel i for the kth hop,

$$x_{ik} = x_i(k\tau); \quad i = 1, 2, \ldots, M; \ k = 1, 2, \ldots, L. \tag{5-5a}$$

Without loss of generality, the signal is assumed present in channel 1. Therefore, the square-law envelope detector outputs on the kth hop $(k = 1, 2, \ldots, L)$ are, when not jammed,

$$\left.\begin{aligned} x_{1k} &= \left(\sqrt{2S}\cos\theta_1 + n_{c1k}\right)^2 + \left(-\sqrt{2S}\sin\theta_1 + n_{s1k}\right)^2 \\ x_{ik} &= n_{cik}^2 + n_{sik}^2; \qquad i = 2, 3, \ldots, M \end{aligned}\right\} \text{with probability } (1-\gamma), \tag{5-5b}$$

and, when jammed,

$$\left.\begin{aligned} x_{1k} &= \left(\sqrt{2S}\cos\theta_1 + n_{c1k} + j_{c1k}\right)^2 \\ &\qquad + \left(-\sqrt{2S}\sin\theta_1 + n_{s1k} + j_{s1k}\right)^2 \\ x_{ik} &= \left(n_{cik} + j_{cik}\right)^2 + \left(n_{sik} + j_{sik}\right)^2; \quad i = 2, 3, \ldots, M \end{aligned}\right\} \text{ with probability } \gamma, \tag{5-5c}$$

where n_{cik}, n_{sik}, $i = 1, 2, \ldots, M$, $k = 1, 2, \ldots, L$, are the independent noise quadrature components in the channels at the sample times $t_k = k\tau$. Thus, x_{1k} is a noncentral chi-squared random variable with two degrees of freedom and x_{ik}, $i = 2, 3, \ldots, M$, are central chi-squared random variables with two degrees of freedom, each scaled by $\sigma_k^2 \equiv \sigma^2(t=k\tau)$.

In the conventional receiver discussed in Section 2, the M detector outputs on the kth hop are linearly combined for all $k = 1, 2, \ldots, L$ to give the decision variables z_i. However, in the present case the decision variables z_i, $i = 1, 2, \ldots, M$, are obtained by first normalizing the detector outputs. Since normalization takes place on a per-hop basis, the resulting weighting is non-uniform and non-linear. The weights W_k are generated by taking the reciprocal of the sum of the sample outputs on a per-hop basis, i.e.

$$W_k = \left(\sum_{i=1}^{M} x_{ik}\right)^{-1}; \quad k = 1, 2, \ldots, L. \tag{5-6}$$

We can now write the weighted variables z_{ik} as

$$z_{ik} = x_{ik}W_k; \quad i = 1, 2, \ldots, M. \tag{5-7a}$$

The decision variables z_i are then obtained by summing the weighted variables z_{ik} for all $k = 1, 2, \ldots, L$. Thus,

$$z_i = \sum_{k=1}^{L} z_{ik}; \quad i = 1, 2, \ldots, M. \tag{5-7b}$$

The symbol decision can now be made on the basis of the largest of the decision variables z_i; $i = 1, 2, \ldots, M$.

In the following subsection, we analyze this model for the general M-ary case to find the probability of symbol error. An example of the general M-ary expression for the probability of error is given following the determination of the joint density function of decision variables for L=1. Due to its complexity (shown by example), we then proceed in subsection 5.3 to analyze a special case for M=2 where the solution for the correlated decision variables may be found for a slightly different form of receiver which is equivalent in performance. An N-level quantizer is considered and the effect of the number of levels on the performance observed.

5.2 PROBABILITY OF ERROR ANALYSIS FOR M-ARY CASE

Using the model described in Section 5.1 we proceed with the analysis to compute the probability of symbol error for the self-normalizing receiver. The symbol error probability, assuming equally likely symbols, is

$$P_s(e;\gamma) = P_s(e;\gamma|m_1) = \sum_{\ell=0}^{L} \Pr(\ell \text{ of } L \text{ hops jammed})\, P_s(e;\gamma|m_1, \ell \text{ hops jammed})$$

$$= \sum_{\ell=0}^{L} \binom{L}{\ell} \gamma^{\ell} (1-\gamma)^{L-\ell} P_s(e;\gamma|m_1,\ell). \tag{5-8}$$

The probability of bit error is obtained using the relation

$$P_b(e) = \frac{M/2}{M-1} P_s(e;\gamma). \tag{5-9}$$

The conditional symbol error probability is more conveniently expressed in terms of the probability of making a correct decision. Thus,

$$P_s(e;\gamma|m_1,\ell) = 1 - P_s(c;\gamma|m_1,\ell), \tag{5-10a}$$

where

$$P_s(c;\gamma|m_1,\ell) = \Pr\{z_2 < z_1,\ z_3 < z_1,\ldots,z_M < z_1\}, \tag{5-10b}$$

This can be expressed as

$$P_s(c;\gamma|m_1,\ell) = \int_0^\infty dz_1 \underbrace{\int_0^{z_1}\int_0^{z_1}\cdots\int_0^{z_1}}_{M-1} p_{z_1 z_2\ldots z_M}(z_1,\ z_2,\ldots,z_M)dz_2 dz_3\ldots dz_M. \tag{5-11}$$

In Sections 2, 3 and 4 we could simplify (5-11) further by noting that the variables z_i are independent. However, in the present case the decision variables z_i, $i = 1, 2,\ldots,M$, are correlated (linearly dependent), since

$$\sum_{i=1}^{M} z_i = L. \tag{5-12}$$

Therefore, the joint pdf of the decision variables is required to determine the error probability.

5.2.1 Joint Density Function of the Normalized Variables

The process of self-normalization has brought about the correlation of the decision variables. The method as was used and described in previous sections of this report assumed that the decision statistics were independent and, therefore, can no longer be applied to this receiver. This subsection derives the general expression of the joint density function of the decision statistics of FH/MFSK receiver for L=1. We assume for our convenience in writing this derivation that f_M is transmitted.

For the single-hop case, we can write the decision variables as

$$z_i = \frac{x_i}{\sum_{i=1}^{M} x_i} \; ; \quad i = 1, 2, \ldots, M, \tag{5-13}$$

where the x_i have the pdf's

$$p_{x_i}(\alpha) = \frac{1}{2\sigma^2} e^{-\alpha/2\sigma^2}, \; x_i \geq 0, \quad i = 1, 2, \ldots, M-1; \tag{5-14a}$$

$$p_{x_M}(\alpha) = \frac{1}{2\sigma^2} e^{-\alpha/2\sigma^2 - \rho} \; I_0\left[\sqrt{2\alpha(\rho/\sigma^2)}\right], \; x_M \geq 0. \tag{5-14b}$$

By letting $y_i = \sum_{k=1}^{i} x_k$, the variables z_i may be expressed by

$$\begin{aligned}
0 \leq z_1 &= y_1/y_M \leq 1 \\
0 \leq z_2 &= (y_2 - y_1)/y_M \leq 1 \\
&\vdots \\
0 \leq z_{M-1} &= (y_{M-1} - y_{M-2})/y_M \leq 1 \\
0 \leq \xi &= y_M < \infty
\end{aligned} \tag{5-15}$$

where $0 \leq y_1 \leq y_2 \leq \ldots \leq y_{M-1} \leq y_M < \infty$. With manipulation of y_i in terms of z_i and ξ we obtain the transformation of variables

$$\left.\begin{aligned}
y_1 &= \xi z_1 \\
y_2 &= \xi(z_1 + z_2) \\
&\vdots \\
y_{M-1} &= \xi \sum_{i=1}^{M-1} z_i \\
y_M &= \xi,
\end{aligned}\right\} \text{ for } y_1 \leq y_2 \leq y_3 \leq \ldots \leq y_M \tag{5-16}$$

with Jacobian

$$J = \xi^{M-1}. \tag{5-17}$$

The pdf of the new variables is

$$p_{\underline{z},\xi}(\xi, z_1, z_2, \ldots, z_{M-1}) = \xi^{M-1}\, p_{\underline{y}}\left[\xi z_1, \xi(z_1 + z_2), \ldots, \xi \sum_{i=1}^{M-1} z_i, \xi\right]. \tag{5-18}$$

Since

$$p_{\underline{y}}(y\ , y\ , \ldots, y_M) = p_{\underline{x}}(y_1, y_2 - y_1, y_3 - y_2, \ldots, y_M - y_{M-1}), \tag{5-19}$$

the pdf in (5-18) can be expressed in terms of x by

$$p_{\underline{z},\xi}(\xi, z_1, z_2, \ldots, z_{M-1}) = \xi^{M-1}\, p_{\underline{x}}\left[\xi z_1, \xi z_2, \ldots, \xi z_{M-1}, \xi\left(1 - \sum_{i=1}^{M-1} z_i\right)\right]. \tag{5-20}$$

Substituting (5-14) into (5-20), we can write

$$p_{\underline{z},\xi}(\xi, z_1, z_2, \ldots, z_{M-1}) = \left(\frac{1}{2\sigma^2}\right)^M \xi^{M-1}\, e^{-\xi/2\sigma^2 - \rho}\, I_0\left[\sqrt{\frac{2\rho}{\sigma^2}\, \xi\left(1 - \sum_{i=1}^{M-1} z_i\right)}\right]. \tag{5-21}$$

Finally, integrating with respect to ξ gives

$$p_{\underline{z}}(z_1, z_2, \ldots, z_{M-1}) = e^{-\rho}\, (M-1)!\ {}_1F_1\left[M;1;\rho\left(1 - \sum_{i=1}^{M-1} z_i\right)\right], \quad 0 < z_i < 1. \tag{5-22}$$

The total joint density function of the decision variables (including z_M) can now be expressed as

$$p_{\underline{z}}(z_1, z_2, \ldots, z_M) = e^{-\rho}\, (M-1)!\ {}_1F_1(M;1;\rho z_M)\, \delta\left(\sum_{i=1}^{M} z_i - 1\right) \tag{5-23}$$

where $\delta(\cdot)$ is the Dirac delta function and

$$\rho = \begin{cases} \rho_N, & \text{with probability } 1-\gamma \\ \rho_T, & \text{with probability } \gamma. \end{cases} \tag{5-24}$$

Equation (5-23) completes the derivation of the general expression of the pdf of the decision statistics of self-normalizing FH/MFSK receiver for the single-hop case.

5.2.2 Error Probability for M-ary Case (L=1)

When L=1, the general expression for the joint density function of the decision variables z_i, $i = 1, 2, \ldots, M$, assuming the signal is present in channel 1, is

$$p_{\underline{z}}(z_1, z_2, \ldots, z_M) = e^{-\rho} (M-1)!\; {}_1F_1(M;1;\rho z_1)\, \delta\left(\sum_{i=1}^{M} z_i - 1\right). \tag{5-25}$$

Using the expression in (5-11), the probability of making a correct decision is

$$P_s(c) = \int_0^1 dz_1 \int_0^{z_1} dz_2 \cdots \int_0^{z_1} dz_M\, e^{-\rho} (M-1)!\; {}_1F_1(M;1;\rho z_1)\, \delta\left(\sum_{i=1}^{M} z_i - 1\right). \tag{5-26}$$

We can further substitute [22, p. 36]

$$\delta\left(\sum_{i=1}^{M} x_i - 1\right) = \frac{1}{2\pi}\int_{-\infty}^{\infty} du\, \exp\left[ju\left(\sum_{i=1}^{M} x_i - 1\right)\right], \tag{5-27}$$

thus giving

$$P_s(c) = \int_0^1 dz_1\, e^{-\rho}(M-1)!\; {}_1F_1(M;1;\rho z_1) \int_0^{z_1} dz_2 \cdots \int_0^{z_1} dz_M\, \frac{1}{2\pi}\int_{-\infty}^{\infty} du\, \exp\left[ju\left(\sum_{i=1}^{M} z_i - 1\right)\right]. \tag{5-28}$$

After rearranging, we have

$$P_s(c) = \int_0^1 dz_1\, e^{-\rho} (M-1)!\; {}_1F_1(M;1;\rho z_1)\, \frac{1}{2\pi}\int_{-\infty}^{\infty} du\, e^{-ju(1-z_1)} \left(\int_0^{z_1} dz_i\, e^{juz_i}\right)^{M-1} \tag{5-29}$$

or

$$P_s(c) = \int_0^1 dz_1\, e^{-\rho}(M-1)!\; {}_1F_1(M;1;\rho z_1)\, \frac{1}{2\pi}\int_{-\infty}^{\infty} du\, e^{-ju(1-z_1)} \left(\frac{e^{juz} - 1}{ju}\right)^{M-1}. \tag{5-30}$$

Equation (5-30) can further be simplified with manipulation of the exponential term in the inner integral. Thus,

$$P_s(c) = \int_0^1 dz_1\, e^{-\rho}\,(M-1)!\,{}_1F_1(M;1;\rho z_1)\,\frac{1}{2\pi}\int_{-\infty}^{\infty} du\, e^{-ju(1-z_1)}\, e^{ju(M-1)z_1/2}$$

$$\cdot \left(\frac{e^{juz_1/2} - e^{-juz_1/2}}{ju}\right)^{M-1} . \qquad (5\text{-}31)$$

Using Euler's identity, we have

$$P_s(c) = \int_0^1 dz_1\, e^{-\rho}\,(M-1)!\,{}_1F_1(M;1;\rho z_1)\,\frac{1}{2\pi}\int_{-\infty}^{\infty} du\, \exp\left\{-ju\left[1 - z_1 - \frac{(M-1)z_1}{2}\right]\right\}$$

$$\cdot \frac{\sin^{M-1}(uz_1/2)}{(u/2)^{M-1}} . \qquad (5\text{-}32)$$

Making the change of variable $w = uz_1$ and simplifying terms, we then have

$$P_s(c) = \int_0^1 dz_1\, e^{-\rho}\,(M-1)!\,{}_1F_1(M;1;\rho z_1)\, z_1^{(M-2)}\,\frac{1}{2\pi}\int_{-\infty}^{\infty} dw\, \exp\left[jw\left(\frac{M+1}{2} - \frac{1}{z_1}\right)\right]\left(\frac{\sin w/2}{w/2}\right)^{M-1} . \qquad (5\text{-}33)$$

The inner integral can be written by its Fourier Transform pair in terms of (M-1) convolutions of a·rectangular function and, therefore, the probability of making a correct decision can be expressed for general M as

$$P_s(c) = \int_0^1 dz_1\, z_1^{M-2}\, e^{-\rho}\,(M-1)!\,{}_1F_1(M;1;\rho z_1)\underbrace{\left[\operatorname{rect}\left(\frac{M+1}{2} - \frac{1}{z_1}\right) * \ldots * \operatorname{rect}\left(\frac{M+1}{2} - \frac{1}{z_1}\right)\right]}_{(M-1)\text{-fold self-convolution}} , \qquad (5\text{-}34a)$$

where the rectangular function is defined by

$$\operatorname{rect}(t) = \begin{cases} 1, & -\frac{1}{2} < t < \frac{1}{2} \\ 0, & \text{elsewhere.} \end{cases} \qquad (5\text{-}34b)$$

To verify this, we show that when we set M=2 in (5-34a) the probability of making a correct decision reduces to

$$P_s(c) = \int_0^1 dz_1\ z_1^{2-2}\ e^{-\rho}\ (2-1)!\ {}_1F_1(2;1;\rho z_1)\ \mathrm{rect}\left(\frac{3}{2} - \frac{1}{z_1}\right), \qquad (5\text{-}35)$$

where $\mathrm{rect}\left(\frac{3}{2} - \frac{1}{z_1}\right)$ is 1 for $-\frac{1}{2} < \frac{3}{2} - \frac{1}{z_1} < \frac{1}{2}$. Thus,

$$\mathrm{rect}\left(\frac{3}{2} - \frac{1}{z_1}\right) = \begin{cases} 1, & \frac{1}{2} < z_1 < 1 \\ 0, & \text{elsewhere,} \end{cases} \qquad (5\text{-}36)$$

which determines the range of integration. This can further be simplified by noting that ${}_1F_1(2;1;\rho z_1) = (1+\rho z_1)\exp(\rho z_1)$. Thus,

$$P_s(c) = \int_{1/2}^{1} dz_1\ e^{-\rho}\ (1+\rho z_1)\ \exp(\rho z_1) \qquad (5\text{-}37)$$

and, after evaluating the integral, (5-37) reduces to the conventional result

$$P_s(c) = 1 - \frac{1}{2} e^{-\rho/2}\ . \qquad (5\text{-}38)$$

For M=4, the probability of correct decision is

$$P_s(c) = \int_0^1 dz_1\ 6z_1^2 e^{-\rho}\, {}_1F_1(4;1;\rho z_1)\ [\mathrm{rect}(x) * \mathrm{rect}(x) * \mathrm{rect}(x)] \qquad (5\text{-}39)$$

where the convolution is to be evaluated with respect to the parameter x defined by.

$$x = \frac{5}{2} - \frac{1}{z_1}\ . \qquad (5\text{-}40)$$

The self-convolution of this rectangular pulse gives

$$f_3(x) = \begin{cases} \frac{1}{2}x^2 + \frac{3}{2}x + \frac{9}{8}, & -\frac{3}{2} < x < -\frac{1}{2} \\ -x^2 + 3/4, & -\frac{1}{2} < x < \frac{1}{2} \\ \frac{1}{2}x^2 + -\frac{3}{2}x + \frac{9}{8}, & \frac{1}{2} < x < \frac{3}{2} \end{cases} \qquad (5\text{-}41a)$$

and after substituting for x, we have

$$f_3(z_1) = \begin{cases} \frac{1}{2}\left(16 - \frac{8}{z_1} + \frac{1}{z_1^2}\right), & \frac{1}{4} < z_1 < \frac{1}{3} \\ \frac{5}{z_1} - \frac{1}{z_1^2} - \frac{11}{2}, & \frac{1}{3} < z_1 < \frac{1}{2} \\ \frac{1}{2}\left(1 - \frac{2}{z_1} + \frac{1}{z_1^2}\right), & \frac{1}{2} < z_1 < 1. \end{cases} \tag{5-41b}$$

When $\rho=0$, the probability of correct decision reduces to

$$P_S(c) = \int_{1/4}^{1/3} 3\left(16z_1^2 - 8z_1 + 1\right) dz_1 + \int_{1/3}^{1/2} \left(30z_1 - 6 - 33z_1^2\right) dz_1$$

$$+ \int_{1/2}^{1} 3\left(z_1^2 - 2z_1 + 1\right) dz_1$$

$$= \frac{1}{4}. \tag{5-42}$$

For general ρ, the probability of correct decision can be expressed, after some algebraic manipulations, as

$$P_S(c) = \int_{1/4}^{1/3} \Big[8\rho^3 z_1^5 + \left(72\rho^2 - 4\rho^3\right) z_1^4 + \left(\frac{1}{2}\rho^3 - 36\rho^2 + 144\rho \; z_1^3\right)$$

$$+ \left(\frac{9}{2}\rho^2 - 72\rho + 48\right)z_1^2 + (9\rho - 24)z_1 + 3\Big] e^{\rho z_1} e^{-\rho} \, dz_1$$

$$\int_{1/3}^{1/2} \Big[- \frac{11}{2}\rho^3 z_1^5 + \left(5\rho^3 - \frac{99}{2}\rho^2\right)z_1^4 + (45\rho^2 - \rho^3 - 99\rho)z_1^3 + (90\rho - 9\rho^2 - 33)z_1^2$$

$$+ (30 - 18\rho)z_1 - 6\Big] e^{\rho z_1} e^{-\rho} dz_1$$

$$+ \int_{1/2}^{1} \left[\frac{1}{2}\rho^3 z_1^5 + \left(\frac{9}{2}\rho^2 - \rho^3\right) z_1^4 + \left(\frac{1}{2}\rho^3 - 9\rho^2 + 9\rho\right) z_1^3 + \left(\frac{9}{2}\rho^2 - 18\rho + 3\right) z_1^2 + (9\rho - 6) z_1 + 3\right] e^{\rho z_1} e^{-\rho} dz_1 \tag{5-43}$$

which reduces to

$$P_s(c) = \left(-\frac{3}{2}e^{\rho/2} + e^{\rho/3} - \frac{1}{4}e^{\rho/4} + e^{\rho}\right)e^{-\rho} . \tag{5-44}$$

The probability of symbol error is, therefore,

$$P_s(e) = e^{-\rho}\left(\frac{3}{2}e^{\rho/2} - e^{\rho/3} + \frac{1}{4}e^{\rho/4}\right) \tag{5-45}$$

which is the same as probability of symbol error for the conventional 4-ary receiver for L=1. This illustrates the fact that the performance of the FH/MFSK self-normalizing receiver is the same as that of the conventional receiver for L=1, since, for this case, each decision variable is the conventional decision variable normalized by same quantity.

5.3 PROBABILITY OF ERROR ANALYSIS FOR BINARY CASE

We have seen that the correlation of the decision variables has greatly complicated the task of obtaining the joint density function of the decision variables. For $M > 2$ and more than one hop/symbol ($L > 1$), it is not evident how to obtain the joint pdf of the weighted and summed variables. In order to find out how well this type of receiver performs in partial-band jamming, we restrict our attention to the binary case (M=2), thus allowing a slight modification of the receiver while maintaining equivalent performance. We also study the effects of including a quantizer and analyze the quantized self-normalizing receiver when the square-law detectors are replaced with linear detectors.

5.3.1 Unquantized Binary System

The system model for the binary case is very similar to the model for the general M-ary case which has been described in Section 5.1. The only difference is that the two normalized detector samples are combined, as shown in Figure 5-2, to obtain a single variable z_k, given by

$$z_k = z_{1k} - z_{2k}. \tag{5-46}$$

For general L, the probability of bit error can be evaluated by

$$P_b(e;\gamma) = \sum_{\ell=0}^{L} \binom{L}{\ell} \gamma^{\ell} (1-\gamma)^{L-\ell} \Pr\left\{\sum_{k=1}^{L} z_k < 0 \,\middle|\, \ell \text{ hops jammed}\right\}. \tag{5-47}$$

The distribution we seek for L-1 is that of

$$z = z_1 - z_2 = \frac{x_{1k} - x_{2k}}{x_{1k} + x_{2k}}, \quad |z| < 1. \tag{5-48}$$

Letting $\alpha = x_{1k}$, $\beta = x_{2k}$, and using the transformation of variables

$$\begin{aligned} z &= (\alpha-\beta)/\alpha+\beta \\ v &= \alpha+\beta \end{aligned} \quad \text{or} \quad \begin{aligned} \alpha &= \tfrac{1}{2} v(1+z) \\ \beta &= \tfrac{1}{2} v(1-z), \end{aligned} \tag{5-49}$$

the joint pdf of z and v has the form

$$\begin{aligned} p_{z,v}(z,v) &= |J|\; p_{\alpha,\beta}(\alpha,\beta) \\ &= \frac{v}{2}\, p_\alpha\left[\frac{1}{2} v(1+z)\right] p_\beta\left[\frac{1}{2} v(1-z)\right], \end{aligned} \tag{5-50}$$

where $p_\alpha(\alpha)$ and $p_\beta(\beta)$ are the density functions defined in (5-14). Integrating with respect to v gives the desired pdf

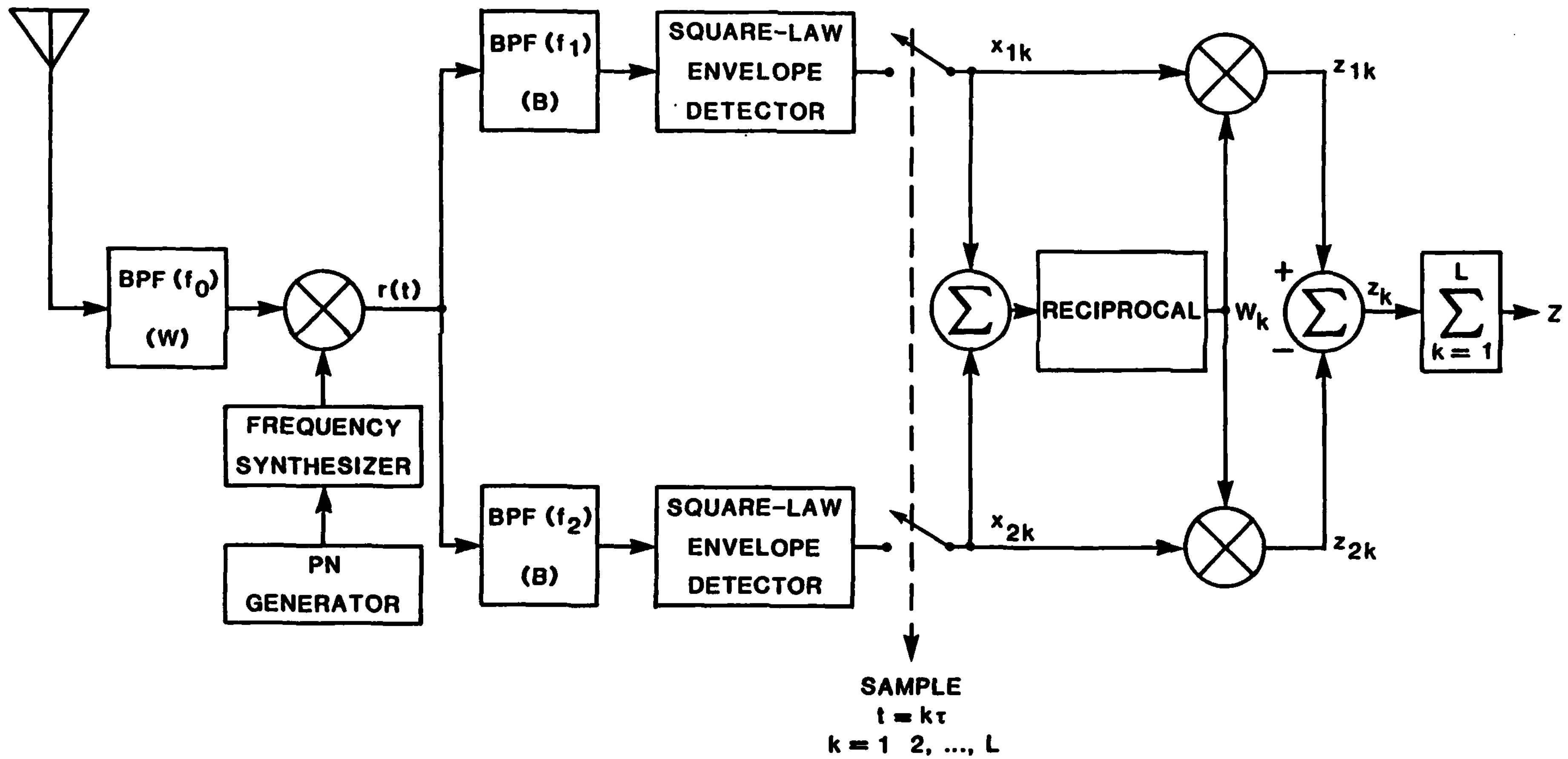


FIGURE 5-2 L HOPS/BIT FH/BFSK SQUARE-LAW COMBINING SELF-NORMALIZING RECEIVER

$$p_z(z) = \int_0^\infty dv\; p_{z,v}(z,v)$$

$$= \int_0^\infty \frac{v}{2}\, p_\alpha\left[\frac{1}{2}\,v(1+z)\right]\, p_\beta\left[\frac{1}{2}\,v(1-z)\right]\, dv. \tag{5-51}$$

The integral in (5-51) can be evaluated to give the desired pdf for L=1:

$$p_z(z;\rho) = \frac{1}{2}\left(1+\frac{\rho}{2}+\frac{\rho z}{2}\right)\, \exp\left[-\frac{\rho}{2}+\frac{\rho z}{2}\right], \quad |z|<1, \tag{5-52}$$

For L=2, $p_z(z;\rho_1,\rho_2)$ can be obtained by direct convolution of (5-52) to give

$$p_z(z;\rho_1,\rho_2) = \begin{cases} \dfrac{1}{2(\rho_1-\rho_2)^3}\Bigg\{e^{-\rho_2+\rho_1 z/2}\,\rho_1\left[-2\rho_2+\rho_1(\rho_1-\rho_2)\left(1+\frac{z}{2}\right)\right] \\ \qquad + e^{-\rho_1+\rho_2 z/2}\,\rho_2\left[2\rho_1+\rho_2(\rho_1-\rho_2)\left(1+\frac{z}{2}\right)\right]\Bigg\}, & -2<z<0 \\ \dfrac{1}{2(\rho_1-\rho_2)^3}\Bigg\{e^{-\rho_2+\rho_2 z/2}\Big[\rho_1^3-\rho_1^2\rho_2-2\rho_1\rho_2 \\ \qquad + (\rho_1-\rho_2)\,(\rho_1^2\rho_2-\rho_1\rho_2^2-\rho_2^2)\,\frac{z}{2}\Big] \\ \qquad - e^{-\rho_1+\rho_1 z/2}\Big[\rho_2^3-\rho_1\rho_2^2-2\rho_1\rho_2 \\ \qquad + (\rho_1-\rho_2)\,(\rho_1^2\rho_2-\rho_1\rho_2^2+\rho_1^2)\,\frac{z}{2}\Big]\Bigg\}, & 0<z<2. \end{cases} \tag{5-53}$$

Thus, the probability of bit error using (5-25) has the form, for L=1,

$$P_b(e;\gamma) = \gamma\frac{1}{2}e^{-\rho_T/2} + (1-\gamma)\frac{1}{2}e^{-\rho_N/2}; \tag{5-54a}$$

and, for L = 2,

$$P_b(e;\gamma) = (1-\gamma)^2 \frac{1}{2} e^{-\rho_N/2}\left(1+\frac{\rho_N}{6}\right) + \gamma^2 \frac{1}{2} e^{-\rho_T/2}\left(1+\frac{\rho_T}{6}\right)$$

$$+ 2\gamma(1-\gamma)\frac{1}{\left(\frac{\rho_N}{2}-\frac{\rho_T}{2}\right)^3}\left\{\frac{1}{2}\left[\frac{1}{2}\rho_N(\rho_N-\rho_T) - (\rho_N+\rho_T)\right]e^{-\rho_T/2}\right.$$

$$\left. + \frac{1}{2}\left[\rho_T(\rho_N-\rho_T) + (\rho_N+\rho_T)\right]e^{-\rho_N/2}\right\} \tag{5-54b}$$

where

$$\rho_N = E_b/N_0;$$

$$\rho_T = \frac{\gamma\left(E_b/N_J\right)\left(E_b/N_0\right)}{\gamma\, E_b/N_J + E_b/N_0}\,. \tag{5-55}$$

For L=3 and L=4, the pdf expressions are obtained by taking the convolution involving (5-45) and (5-46). The final error probability expressions are, however, quite cumbersome and involve many terms. Thus, a numerical approach has been used. A computer listing for the calculation of the error probabilities derived in this subsection is given in Appendix 5A for general L, up to 4. The equations used in this program are given in Appendix 5B.

5.3.2 Binary System Employing Square-Law Detector and Quantizer

The system under consideration has the configuration illustrated in Figure 5-3. The only difference between this receiver structure and the receiver illustrated in Figure 5-2 is that the discrete N-level quantizer, depicted in Figure 5-4, is inserted before the accumulator. The threshold, η, may vary from 0 to ∞; when $\eta = 0$, the N-level quantization becomes a two-level quantizer (hard-limiter).

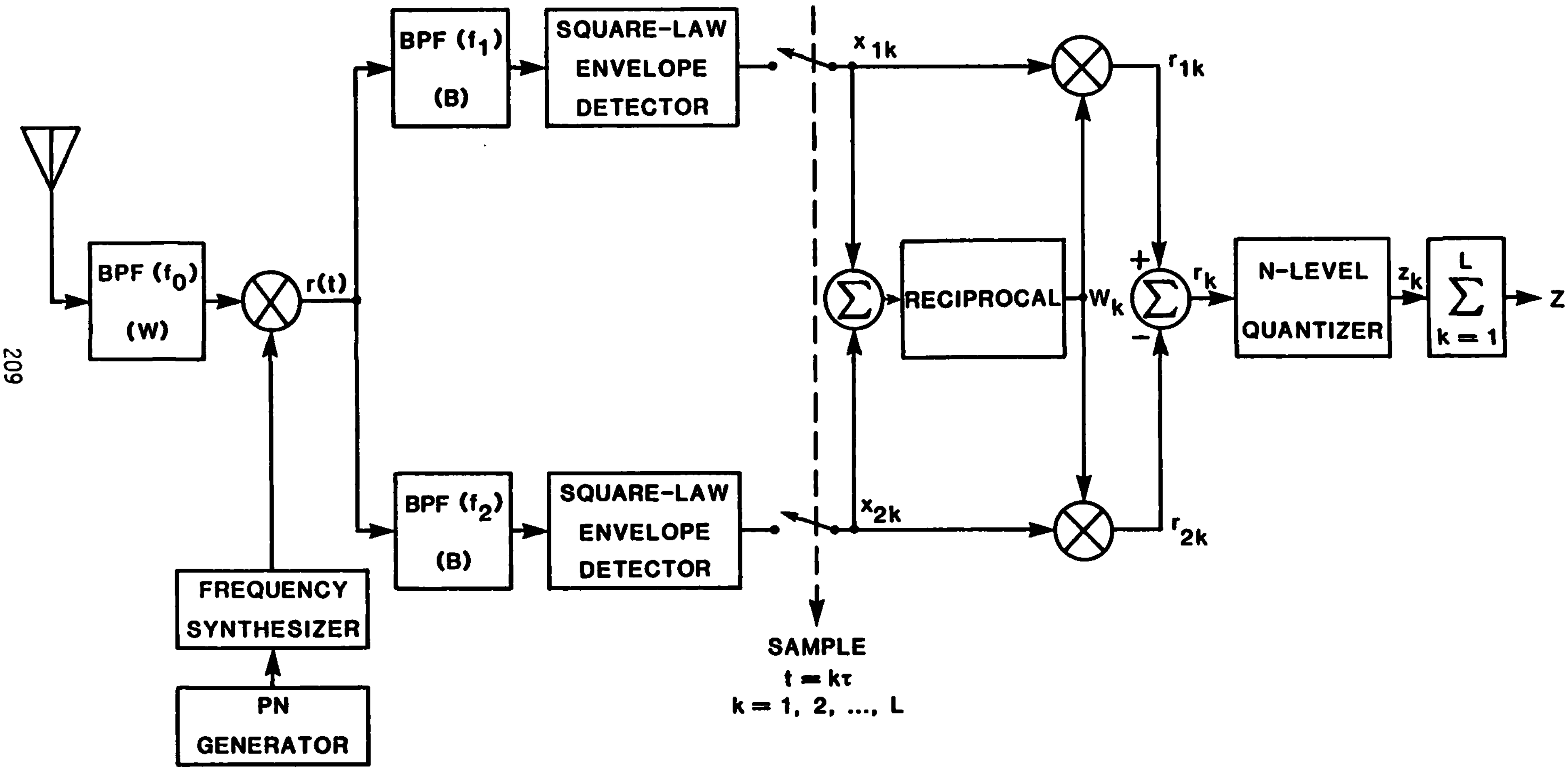


FIGURE 5-3 L HOPS/BIT FH/BFSK SQUARE-LAW COMBINING SELF-NORMALIZING RECEIVER WITH QUANTIZER

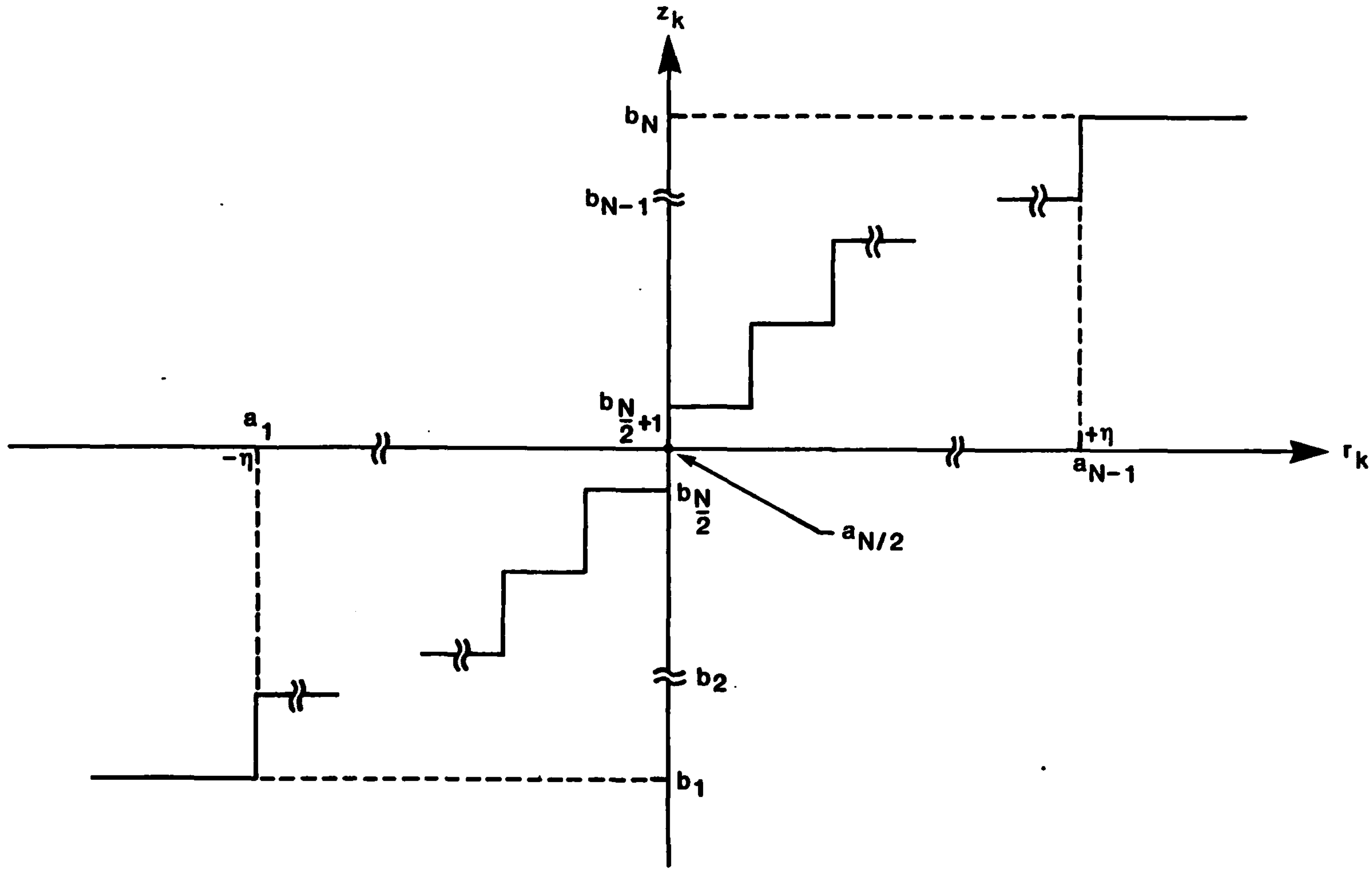


FIGURE 5-4 N-LEVEL QUANTIZER WITH INPUT r_k AND OUTPUT z_k ($a_0 = -\infty$, $a_N = +\infty$, $a_{N/2} = 0$)

The quantizer serves a dual purpose. First, it provides a stepping stone towards a digital system, in which case the number of quantization levels needed for minimizing the quantization error can be evaluated. Second, it will be shown that, for L greater than one, there is an optimum threshold, η, associated with N-level quantization; this optimum threshold minimizes the quantization error and permits the performance of an unquantized receiver to be approximated by that of one with a quantizer.

The input to the quantizer is a set of random variables $\{r_k\}$, where

$$r_k = \frac{x_{1k} - x_{2k}}{x_{1k} + x_{2k}}\,; \quad k = 1, 2, \ldots, L. \tag{5-56}$$

It is quite obvious that r_k may only have values between -1 and +1. The pdf of r_k has been evaluated in (5-52) and is given by

$$p_{r_k}(u,\rho_k) = \begin{cases} \frac{1}{2}\left(1 + \frac{\rho_k}{2} + \frac{\rho_k}{2}\,u\right) \exp\left(-\frac{\rho_k}{2} + \frac{\rho_k}{2}\,u\right), & |u| < 1; \\ 0, & \text{elsewhere.} \end{cases} \tag{5-57}$$

The characteristic of the quantizer is given by

$$a_{i-1} < r_k < a_i \Longleftrightarrow z_k = b_i, \quad i = 1, 2, \ldots, N, \tag{5-58a}$$

where

$$a_0 = -\infty \tag{5-58b}$$

and

$$a_N = \infty. \tag{5-58c}$$

We may define the discrete probabilities

$$\begin{aligned} V_i &= \Pr\left\{z_k = b_i\right\} \\ &= \Pr\left\{a_{i-1} < r_k < a_i\right\}, \qquad i = 1, 2, \ldots, N. \end{aligned} \tag{5-59}$$

Since r_k is limited by [-1,1], the discrete probabilities can then be expressed as

$$V_i = \begin{cases} \Pr\{a_{i-1} < r_k < a_i\}, & a_{i-1} > -1 \text{ and } a_i < 1 \\ \Pr\{-1 < r_k < a_i\}, & a_{i-1} < -1 \text{ and } a_i < 1 \\ \Pr\{a_{i-1} < r_k < 1\}, & a_{i-1} > -1 \text{ and } a_i > 1 \\ \Pr\{-1 < r_k < 1\}, & a_{i-1} < -1 \text{ and } a_i > 1; \end{cases}$$

$$i = 2, 3, \ldots, N-1, \qquad (5\text{-}60a)$$

or, equivalently,

$$V_i = \Pr\{\max(a_{i-1},-1) < r_k < \min(a_i,1)\}; \qquad i = 2, 3, \ldots, N-1, \quad (5\text{-}60b)$$

and V_1 and V_N can be separately determined. There are two separate cases to consider: *(1) when the threshold is less than one and (2) when the threshold* is greater than one. It is obvious that when the threshold η is greater or equal to one, the discrete probabilities V_1 and V_N are 0, since the random variables r_k do not take values in the ranges (η,∞) and $(-\infty,-\eta)$.

Using the density function of r_k given in (5-57), we may write

$$V_i = \begin{cases} \displaystyle\int_{\max(a_{i-1},-1)}^{\min(a_i,1)} \frac{1}{2}\left(1+\frac{\rho_k}{2}+\frac{\rho_k}{2}u\right)\exp\left(-\frac{\rho_k}{2}+\frac{\rho_k}{2}u\right)du, & i = 2, 3, \ldots, N-1 \\ \left.\begin{array}{l} \displaystyle\int_{-1}^{-\eta} \frac{1}{2}\left(1+\frac{\rho_k}{2}+\frac{\rho_k}{2}u\right)\exp\left(-\frac{\rho_k}{2}+\frac{\rho_k}{2}u\right)du, \quad i = 1 \\ \displaystyle\int_{\eta}^{1} \frac{1}{2}\left(1+\frac{\rho_k}{2}+\frac{\rho_k}{2}u\right)\exp\left(-\frac{\rho_k}{2}+\frac{\rho_k}{2}u\right)du, \quad i = N \end{array}\right\} \eta < 1 \\ \left.\begin{array}{ll} 0, & i = 1 \\ 0, & i = N \end{array}\right\} \text{where } \eta \geq 1 \end{cases} \tag{5-61a}$$

or

$$V_i = \begin{cases} \dfrac{1}{2} e^{-\rho_k/2}\left\{\left[1+\min\left(1,a_i\right)\right] e^{\frac{\rho_k}{2}\min\left(1,a_i\right)} - \left[1+\max\left(-1,a_{i-1}\right)\right] e^{\frac{\rho_k}{2}\max\left(-1,a_{i-1}\right)}\right\}, & i = 2, 3, \ldots, N-1 \\ \dfrac{1}{2} e^{-\rho_k/2}\left[(1-\eta)e^{\frac{\rho_k}{2}\eta}\right], & i = 1,\ \eta < 1 \\ 1-\dfrac{1}{2}(1+\eta)\, e^{\frac{\rho_k}{2}(\eta-1)}, & i = N,\ \eta < 1 \\ 0, & i = 1 \end{cases} \tag{5-61b}$$

From the quantizer model, the step size is given by

$$q = \frac{2\eta}{N-2} \tag{5-62}$$

and a_i is given by

$$a_i = \left(-\frac{N}{2} + i\right)q, \quad i = 1, 2, \ldots, N-1. \tag{5-63}$$

Therefore a_i may be expressed in terms of η as

$$a_i = \left(-\frac{N}{2} + i\right)\frac{2\eta}{N-2}, \; i = 1, 2, \ldots, N-1, \tag{5-64}$$

which is a function of threshold η and level of quantization N. Having found the discrete probabilities V_i, i = 1, 2,...,N, we may express the probability density function of z_k as

$$p_{z_k}(\alpha) = \sum_{i=1}^{L(N-1)+1} V_i^{(L)} \, \delta(\alpha - b_i) \tag{5-65}$$

where $V_i^{(L)}$ is the L-fold convolution of V_i, which can be obtained iteratively by*

$$V_k^{(L)} = \sum_{\substack{i=1 \\ i+j-1=k}}^{N} \sum_{j=1}^{(L-1)(N-1)+1} V_i \, V_j^{(L-1)}. \tag{5-66}$$

For the no-jamming case (J_e=0), $V_k^{(L)}$ is the L-fold convolution of V_i in (5-61) with all $\rho_k = \rho_N = S/\sigma_N^2$. Under this assumption, the optimum threshold is obtained. The usual method to determine the optimum threshold that gives the minimum error probability is to differentiate the error expression with respect to η, set the result equal to zero, and find the root of the resulting equation. The probability of error expression for the no-jamming case is

*See Appendix 5F for an alternate formulation.

$$P_b(e) = \begin{cases} \displaystyle\sum_{k=1}^{\frac{L(N-1)}{2}} V_k^{(L)} + \frac{1}{2} V_{\frac{L(N-1)}{2}+1} & \text{for } L(N-1) = \text{even} \\ \displaystyle\sum_{k=1}^{\frac{L(N-1)+1}{2}} V_k^{(L)} & \text{for } L(N-1) = \text{odd.} \end{cases} \tag{5-67}$$

When N is even (e.g., N=4, 8, 16, 32, 64, 128, 256, etc.), the probability of error for the no-jamming case is

$$P_b(e) = \begin{cases} \displaystyle\sum_{k=1}^{\frac{L(N-1)}{2}} V_k^{(L)} + \frac{1}{2} V_{\frac{L(N-1)}{2}+1} & \text{for } L \text{ even} \\ \displaystyle\sum_{k=1}^{\frac{L(N-1)+1}{2}} V_k^{(L)} & \text{for } L \text{ odd .} \end{cases} \tag{5-68}$$

Since (5-68) is too complicated to differentiate with respect to η and solve the equation $dP_b(e)/d\eta = 0$, the optimization is done numerically by searching for the minimum error probability while varying η. This is the most appropriate choice since the analytical approach gets more involved and tedious when L is large. When L=1 the analytical approach shows that the error probability is independent of η and this is confirmed numerically. Having obtained the set of optimum thresholds for different values of L and N, we can proceed with the calculation of error performance under optimum partial-band jamming.

Under partial-band noise jamming, the conditional error probability for ℓ hops jammed and $(L-\ell)$ hops unjammed takes the form of equation (5-68) where the $V_k^{(L)}$ are obtained by the L-fold convolution of V_i given in (5-61), wherein the L-fold convolution is obtained as the ℓ-fold convolution

of V_i with $\rho_k = \rho_T$ convolved with the remaining $(L-\ell)$-fold convolution of V_i with $\rho_k = \rho_N$. The unconditional probability of error is then obtained by averaging the conditional error probability over the possible jamming events. Thus

$$P_b(e) = \sum_{\ell=0}^{L} p_\ell \cdot P_b(e|\ell \text{ hops jammed}) \tag{5-69a}$$

or

$$P_b(e) = \sum_{\ell=0}^{L} \binom{L}{\ell} \gamma^\ell (1-\gamma)^{L-\ell} P_b(e|\ell \text{ hops jammed}) \tag{5-69b}$$

since

$$p_\ell = \binom{L}{\ell} \gamma^\ell (1-\gamma)^{L-\ell}. \tag{5-69c}$$

A listing of the computer program to perform the calculations defined in this subsection is given in Appendix 5C.

5.3.3 Binary System Employing Linear-Law Detector and Quantizer

In Section 5.3.2 we analyzed the system given in Figure 5-3 where the N-level quantizer was the point of discussion. This subsection will also deal with an N-level quantizer and the receiver structure will be identical to Figure 5-3 with only one exception: the squarers following the envelope detectors are removed. This receiver structure is shown in Figure 5-5.

Since the difference between the receiver with square-law detector and one with linear detector lies only in the pdf of the detector outputs, we can avoid redundancy in our analysis by simply stating the differences rather than repeating the derivation which has already been done in Section 5.3.2.

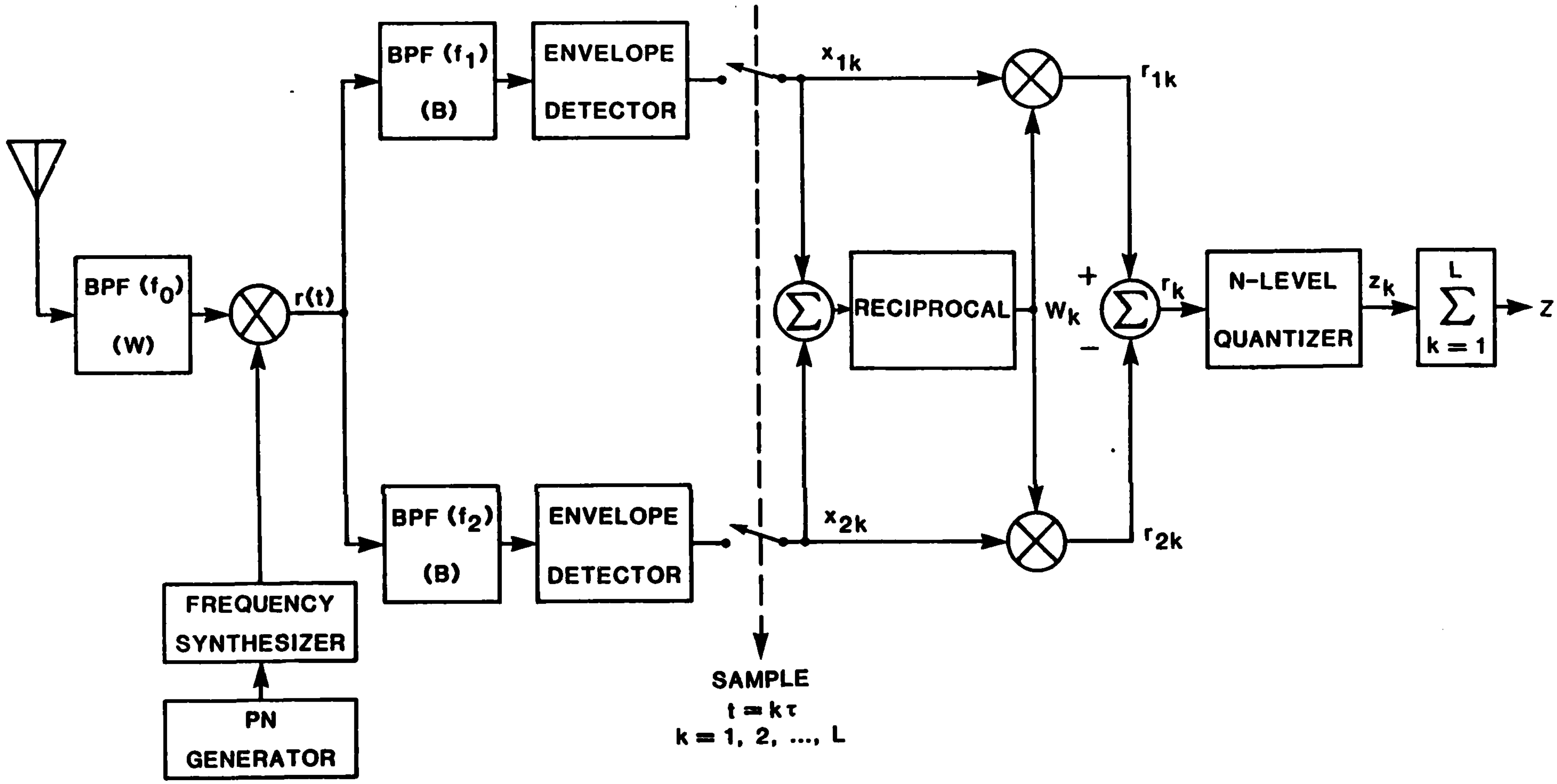


FIGURE 5-5 L HOPS/BIT FH/BFSK LINEAR-LAW COMBINING SELF-NORMALIZING RECEIVER WITH QUANTIZER

We begin with the pdf of r_k, which is derived in Appendix 5D, given by (compared with (5-57) for the square-law detector)

$$p_{r_k}(u,\rho) = \begin{cases} \dfrac{(1-u^2)}{(1+u^2)^2}\left[1+\rho\,\dfrac{(1+u)^2}{2(1+u^2)}\right] \exp\left\{-\rho\left[\dfrac{(1-u)^2}{2(1+u^2)}\right]\right\}, |u| < 1 \\ 0, \text{ elsewhere.} \end{cases} \tag{5-70}$$

The discrete probabilities V_i, $i = 1, 2, \ldots, N$, can be expressed as

$$V_i = \begin{cases} e^{-\rho A}(-A+1) - e^{-\rho B}(-B+1), \quad i = 2, 3, \ldots, N-1 \\ \left.\begin{array}{l} \exp\left[-\rho\,\dfrac{(1+\eta)^2}{2(1+\eta^2)}\right]\left[1-\dfrac{(1+\eta)^2}{2(1+\eta^2)}\right], \quad i=1 \\ 1-\exp\left[-\rho\,\dfrac{(1-\eta)^2}{2(1+\eta^2)}\right]\left[1-\dfrac{(1-\eta)^2}{2(1+\eta^2)}\right], i=N \end{array}\right\} \eta < 1 \\ 0, \quad i=1 \text{ or } i=N \text{, and } \eta \geq 1 \end{cases} \tag{5-71a}$$

where $A = \left[1-\min(a_i,1)\right]^2 \Big/ \left\{2\left[1+\min^2(a_i,1)\right]\right\}$ (5-71b)

$$B = \left[1-\max(a_{i-1},-1)\right]^2 \Big/ \left\{2\left[1+\max^2(a_{i-1},-1)\right]\right\} \tag{5-71c}$$

and

$$a_i = \left(-\frac{N}{2}+i\right)\frac{2\eta}{N-2}\,. \tag{5-71d}$$

From this point on, the analysis follows the procedures in Section 5.3.2 starting with (5-62) and continuing through (5-69). Since the difference between the square-law and linear-law detector is embedded in the discrete probabilities V_i, $i = 1, 2, \ldots, N$, the only modification involved in the computer program is to replace the subroutine VALUE in the program contained in Appendix 5C (listing page 10) with the subroutine given in Appendix 5E.

5.4 NUMERICAL RESULTS FOR THE SELF-NORMALIZING RECEIVER

In this subsection, the performance of the self-normalizing receivers are presented for both wideband noise jamming (γ=1) and optimum partial-band jamming with E_b/N_0 and E_b/N_J as parameters. We have selected practical values of E_b/N_0 such as 13.35 dB, 12.31 dB and 10.95 dB, for which the probability of bit error becomes 10^{-5}, 10^{-4}, and 10^{-3} respectively, under jamming-free conditions for L=1 (i.e., no combining loss).

In the previous sections, we have obtained the expressions for the probability of bit error of the L-hops/symbol self-normalizing FH/BFSK receiver as a function of the jamming fraction, γ. The most effective jamming strategy is to distribute the total jamming power J (i.e., choose γ) in such a way as to cause the communicator to have maximum probability of error. We will denote this optimum value of γ by the symbol γ_0. The usual method to determine the optimum fraction of the band is to differentiate the error probability expression with respect to γ, set the result equal to zero, and find the root of the resulting equation. The optimum γ is the solution to the equation

$$\left.\frac{d\,P(e;\gamma)}{d\gamma}\right|_{\gamma=\gamma_0} = 0. \tag{5-72}$$

However, this approach is abandoned since the analytical solution leads to a formidable task with increasing complexity when L is increased. The only practical method is a numerical and/or graphical search for the maximum error probability as a function of γ.

5.4.1 Unquantized Self-Normalizing Receiver

Figures 5-6 through 5-9 show $P_b(e)$ as a function of γ for E_b/N_0 = 13.35 dB and L=1, 2, 3, and 4, with E_b/N_J as parameter for M=2. As seen, the error probabilities are unimodal functions of γ. Observation of these unimodal curves

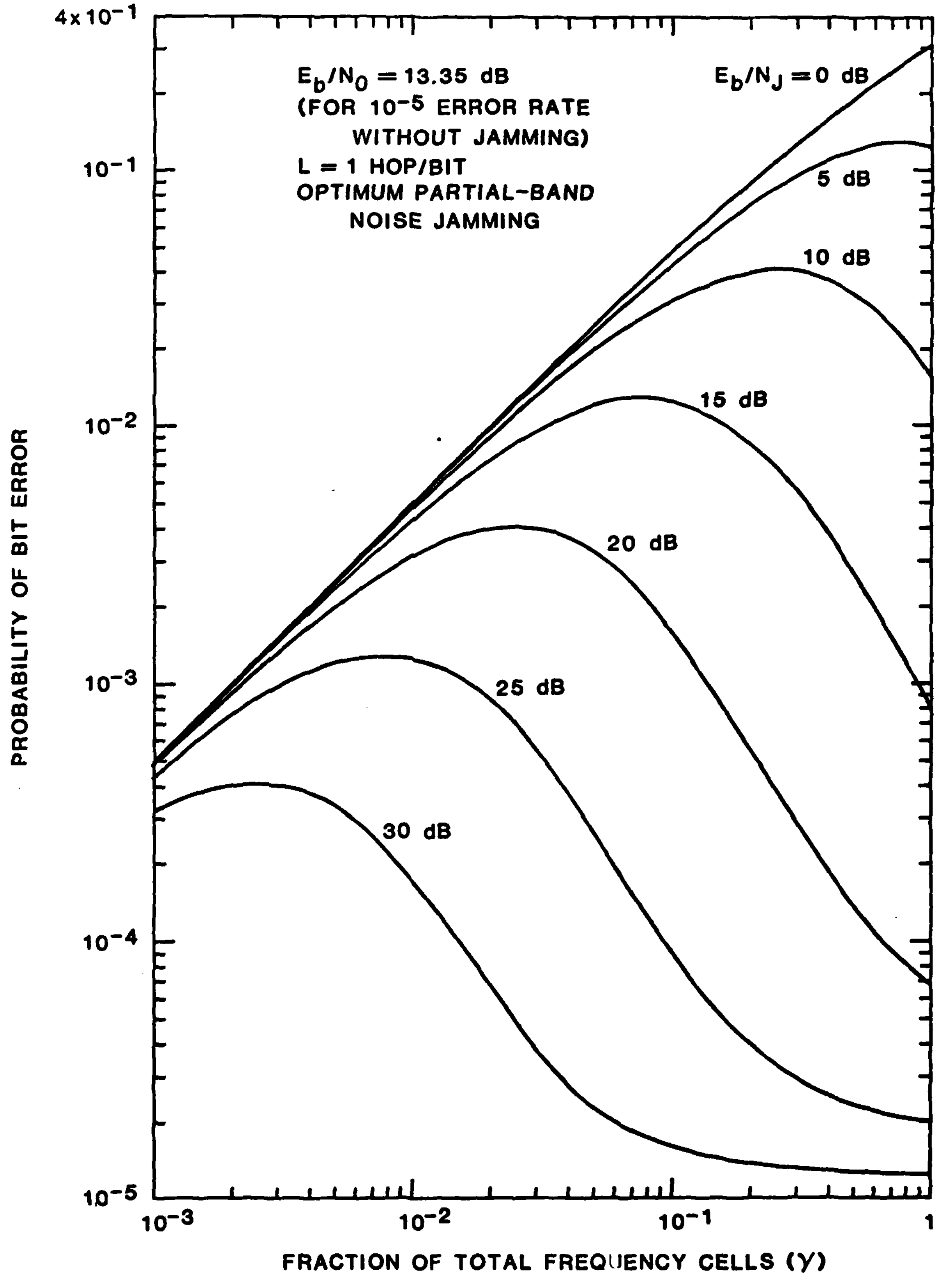


FIGURE 5-6 PROBABILITY OF BIT ERROR VS. FRACTION OF TOTAL FREQUENCY CELLS JAMMED FOR FH/BFSK SELF-NORMALIZING RECEIVER WITH L = 1 HOP/BIT WHEN $E_b/N_0 = 13.35$ dB

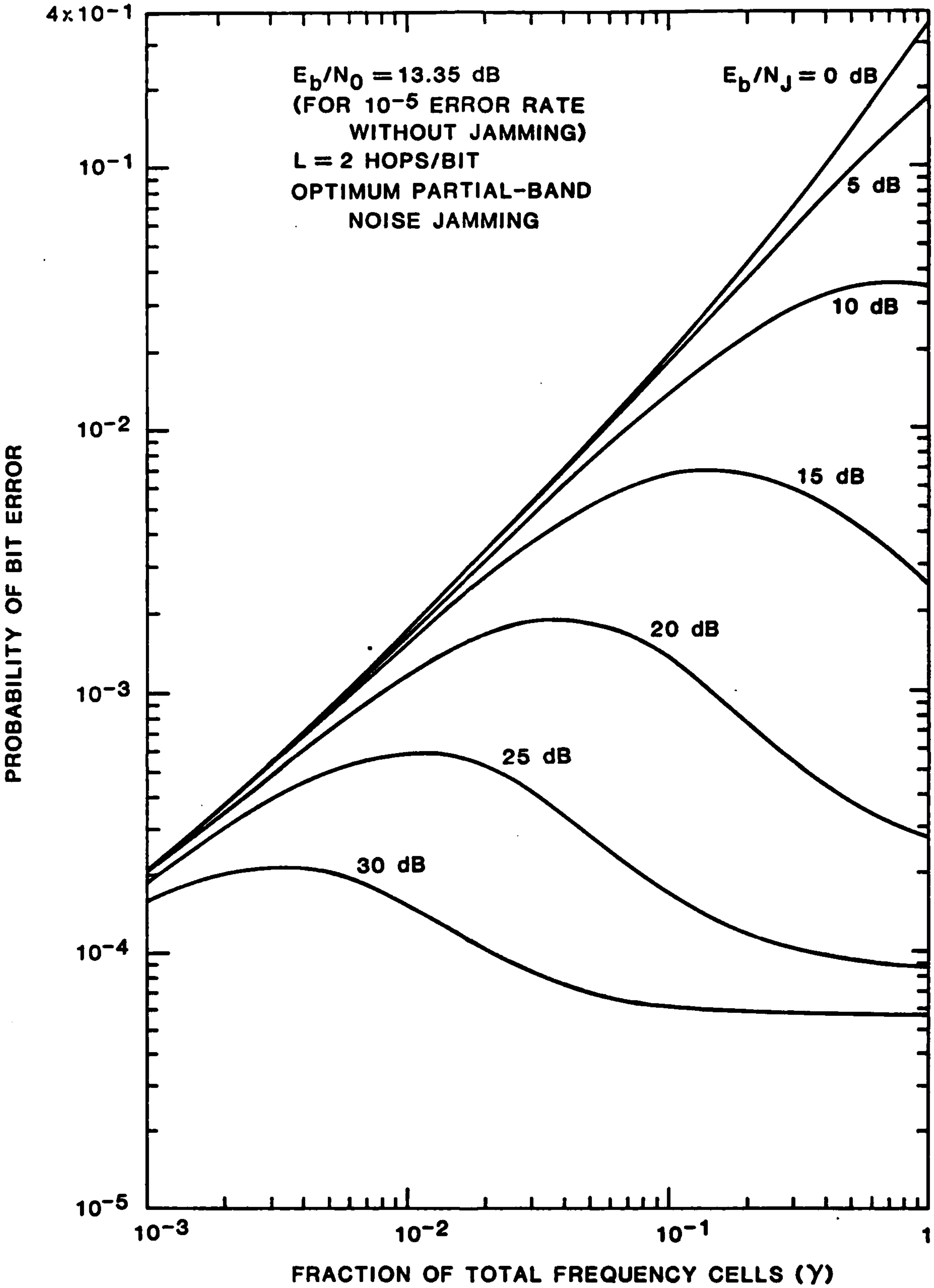


FIGURE 5-7 PROBABILITY OF BIT ERROR VS. FRACTION OF TOTAL FREQUENCY CELLS JAMMED FOR FH/BFSK SELF-NORMALIZING RECEIVER WITH L = 2 HOPS/BIT WHEN $E_b/N_0 = 13.35$ dB

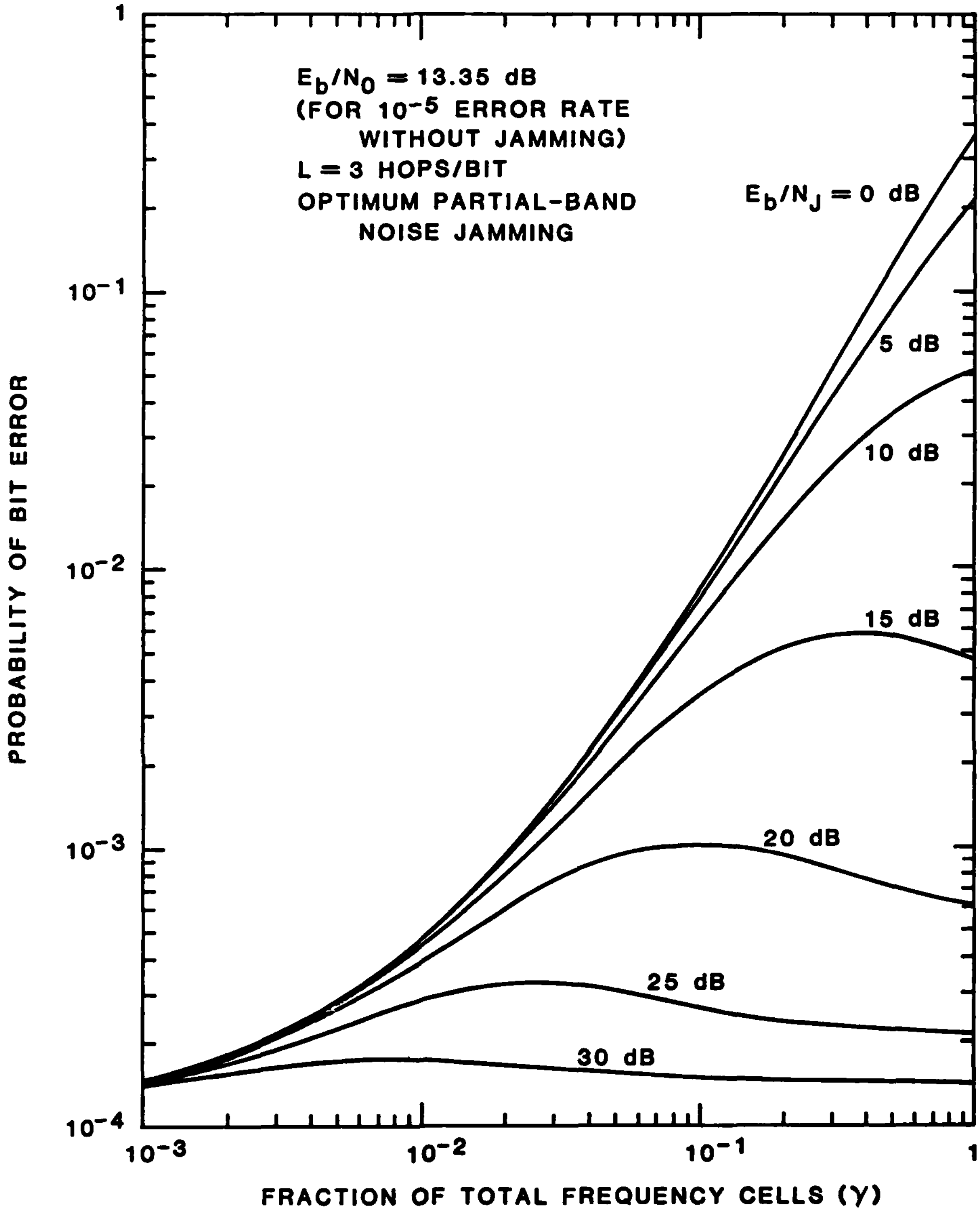


FIGURE 5-8 PROBABILITY OF BIT ERROR VS. FRACTION OF TOTAL FREQUENCY CELLS JAMMED FOR FH/BFSK SELF-NORMALIZING RECEIVER WITH L = 3 HOPS/BIT WHEN $E_b/N_0 = 13.35$ dB

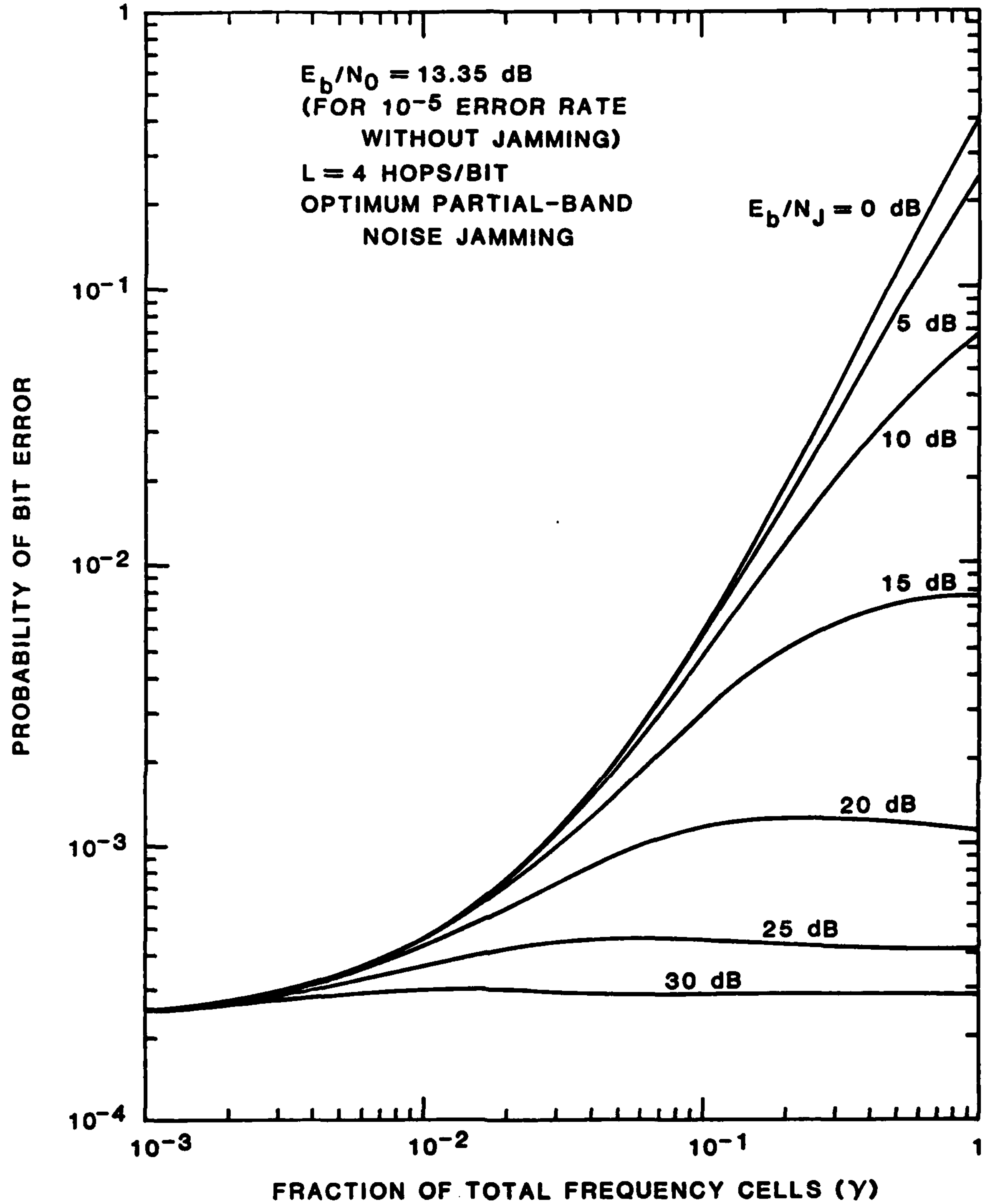


FIGURE 5-9 PROBABILITY OF BIT ERROR VS. FRACTION OF TOTAL FREQUENCY CELLS JAMMED FOR FH/BFSK SELF-NORMALIZING RECEIVER WITH L = 4 HOPS/BIT WHEN $E_b/N_0 = 13.35$ dB

shows that the optimum γ decreases with increasing E_b/N_J. However, as we can see, the optimum γ is brought closer to 1 when L is increased.

Figures 5-10 through 5-12 show the maximum error probabilities corresponding to the optimum γ with L as a parameter for $E_b/N_0 = 13.35$ dB, 12.31 dB, and 10.94 dB, respectively. From these figures, it is seen that the cross-over behavior of the curves for different L under optimum jamming is strongly influenced by the value of E_b/N_0. For example, in Figure 5-12 no cross-over takes place, but as E_b/N_0 is increased, the cross-over behavior becomes more pronounced. This is the same quasi-diversity behavior which occurs for the clipper and AGC receivers discussed in Sections 3 and 4.

In Figures 5-13 and 5-15, we show the wideband jamming performances of the self-normalizing receiver with L as a parameter for $E_b/N_0 = 13.35$ dB, 12.31 dB, and 10.94 dB, respectively. The figures clearly show that as L is increased, the performance degrades due to the noncoherent combining loss. When E_b/N_J becomes very high, the optimum jamming (Figures 5-10 through 5-12) and wideband jamming (Figures 5-13 through 5-15) performances for the self-normalizing receiver approach the same asymptotic values for each L.

5.4.2 Quantized Self-Normalizing Receivers

The effect of the number of quantization levels, N, on the optimum threshold is shown in Figure 5-16 as a function of L. When L=1, the optimized threshold is independent of the quantization level, which is expected, since the quantization is applied at the output of the difference of the signal and noise channels and, thus, no matter what threshold is used, the sum of all V_i, $i = 1, 2, \ldots, N/2$, remains the same. When N increases, the optimum threshold also increases and approaches 1 asymptotically. The plot of optimum threshold vs. the number of quantization levels for a linear-law FH/BFSK self-normalizing receiver is also shown in Figure 5-16 for comparison purposes.

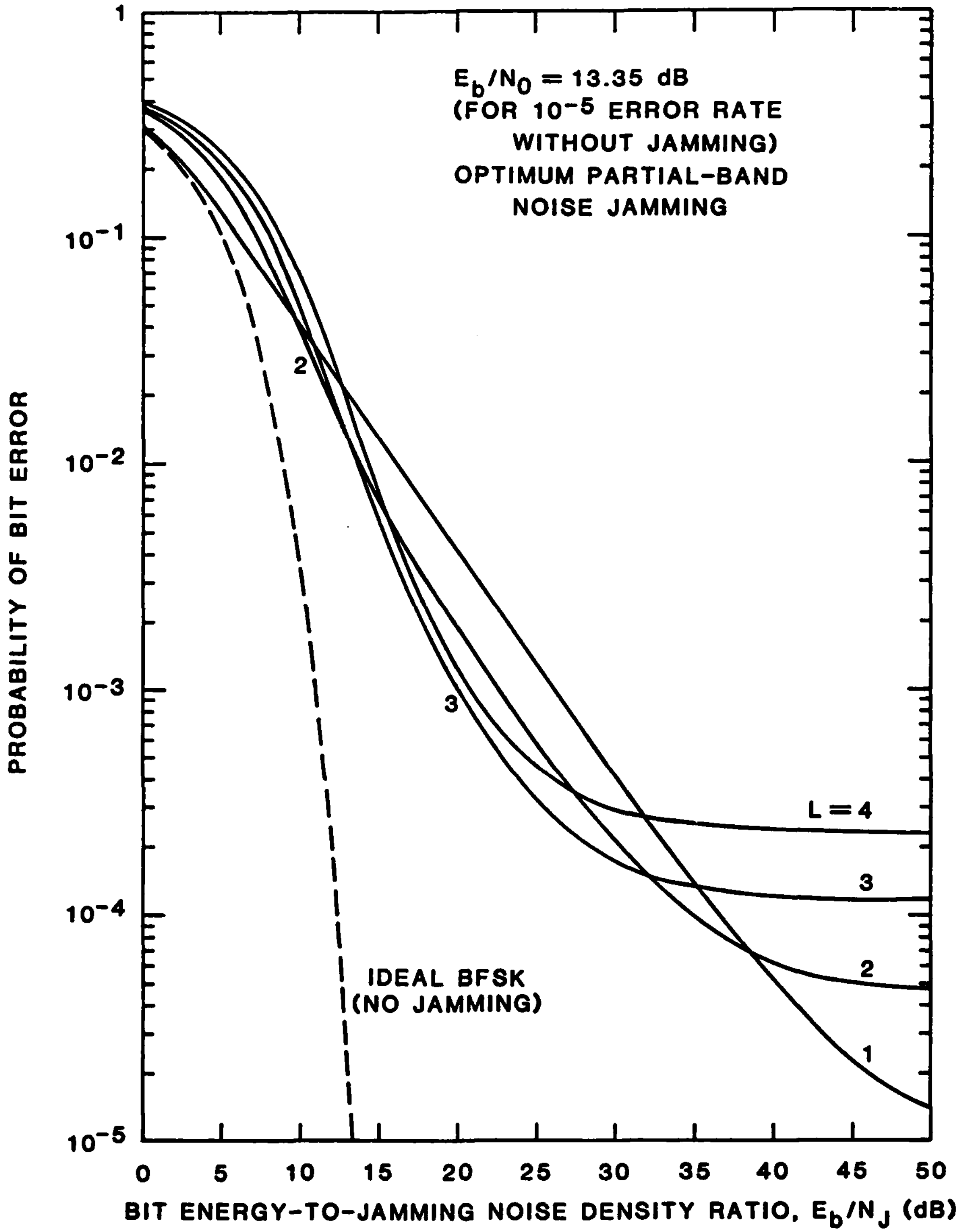


FIGURE 5-10 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/BFSK SELF-NORMALIZING RECEIVER WHEN $E_b/N_0 = 13.35$ dB WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER

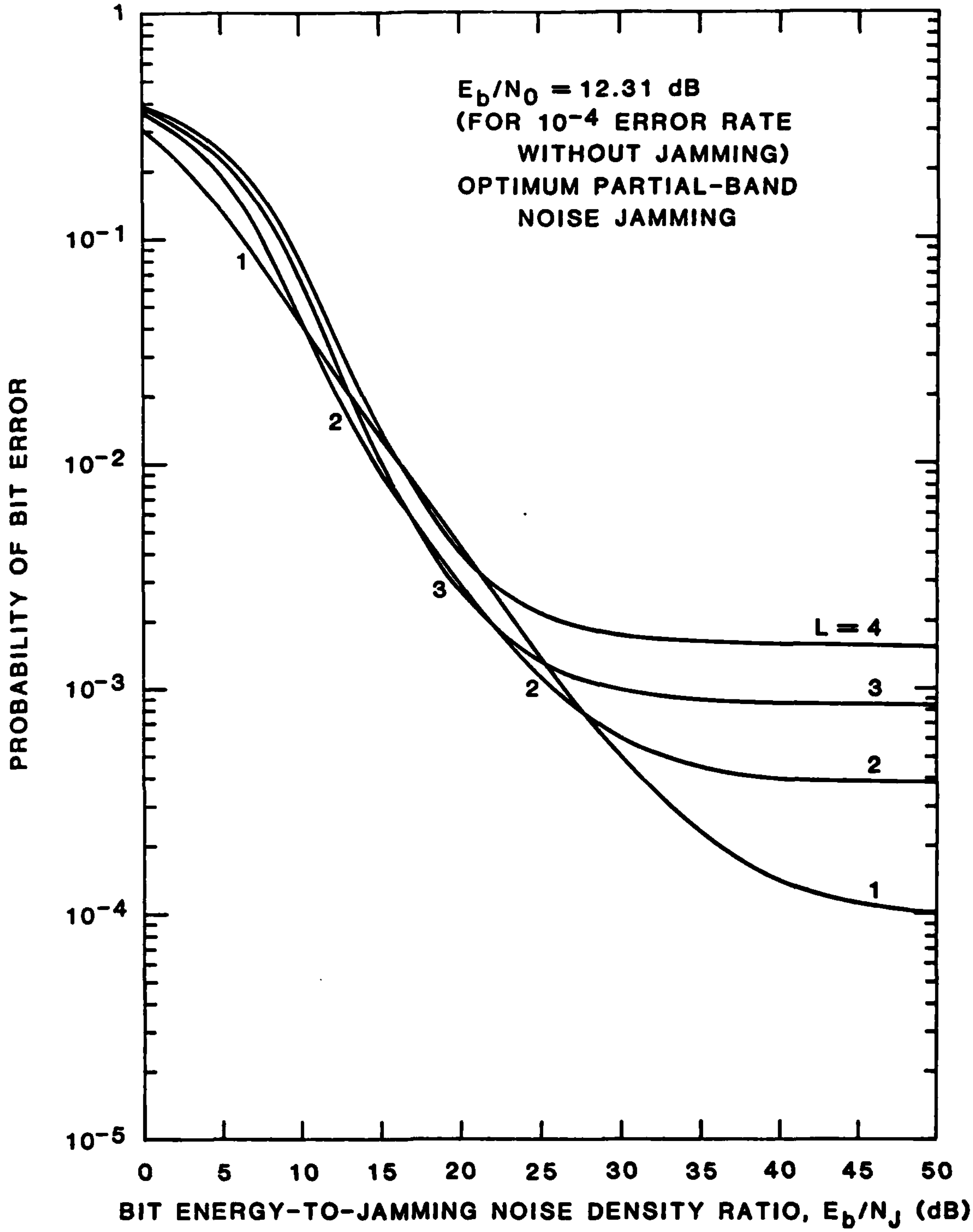


FIGURE 5-11 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/BFSK SELF-NORMALIZING RECEIVER WHEN $E_b/N_0 = 12.31$ dB WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER

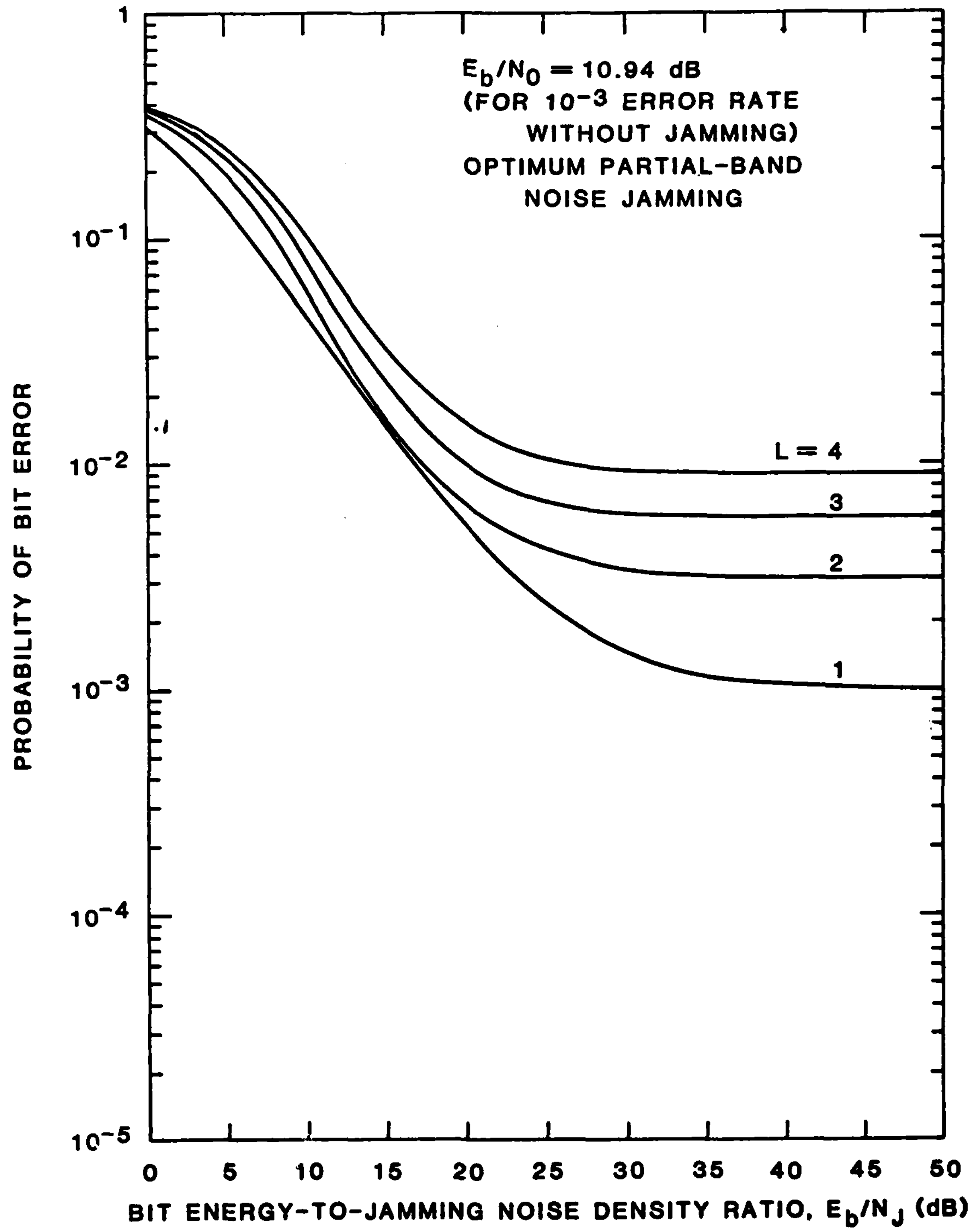


FIGURE 5-12 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/BFSK SELF-NORMALIZING RECEIVER WHEN $E_b/N_0 = 10.94$ dB WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER

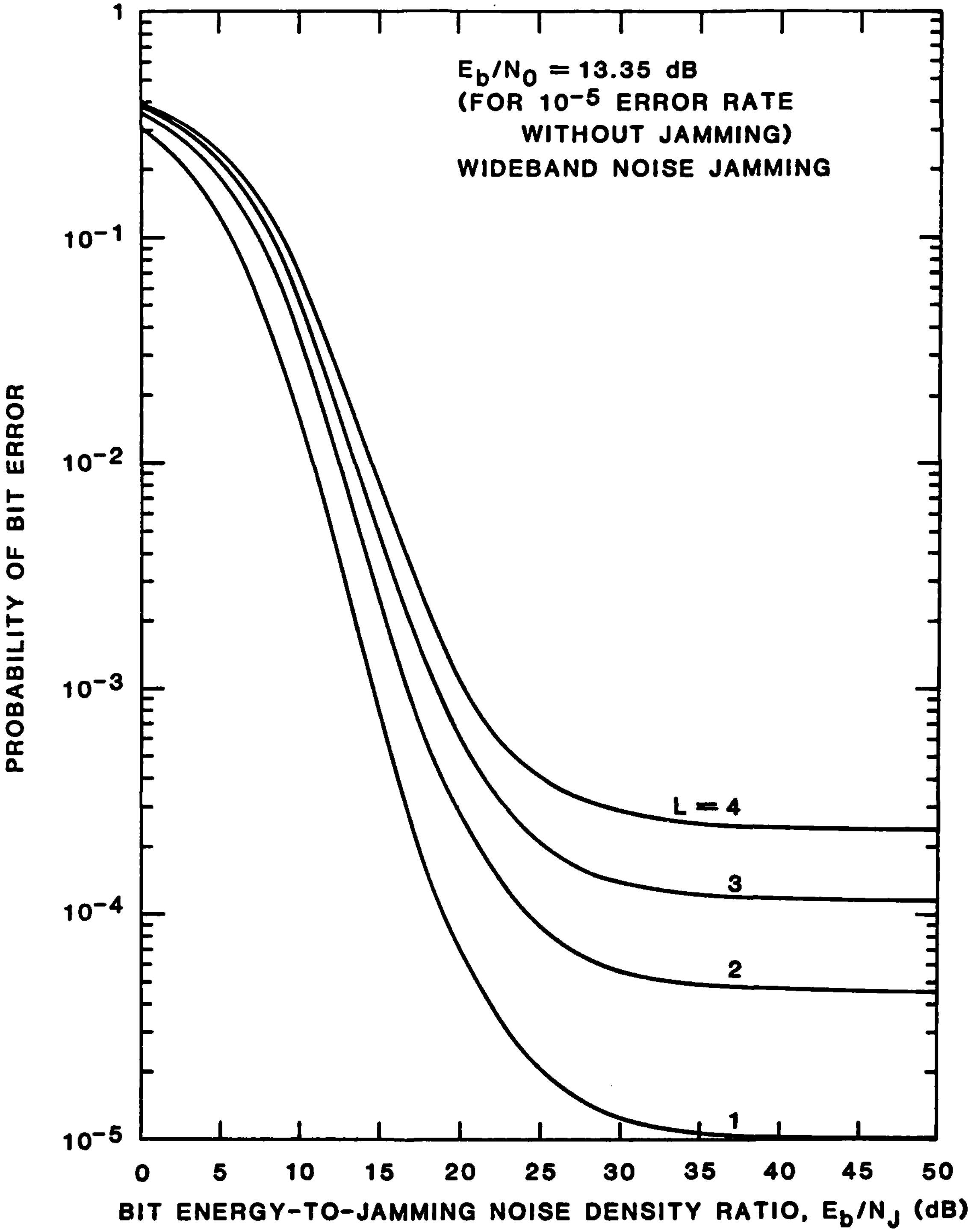


FIGURE 5-13 WIDEBAND NOISE JAMMING PERFORMANCE OF FH/BFSK SELF-NORMALIZING RECEIVER WHEN $E_b/N_0 = 13.35$ dB WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER

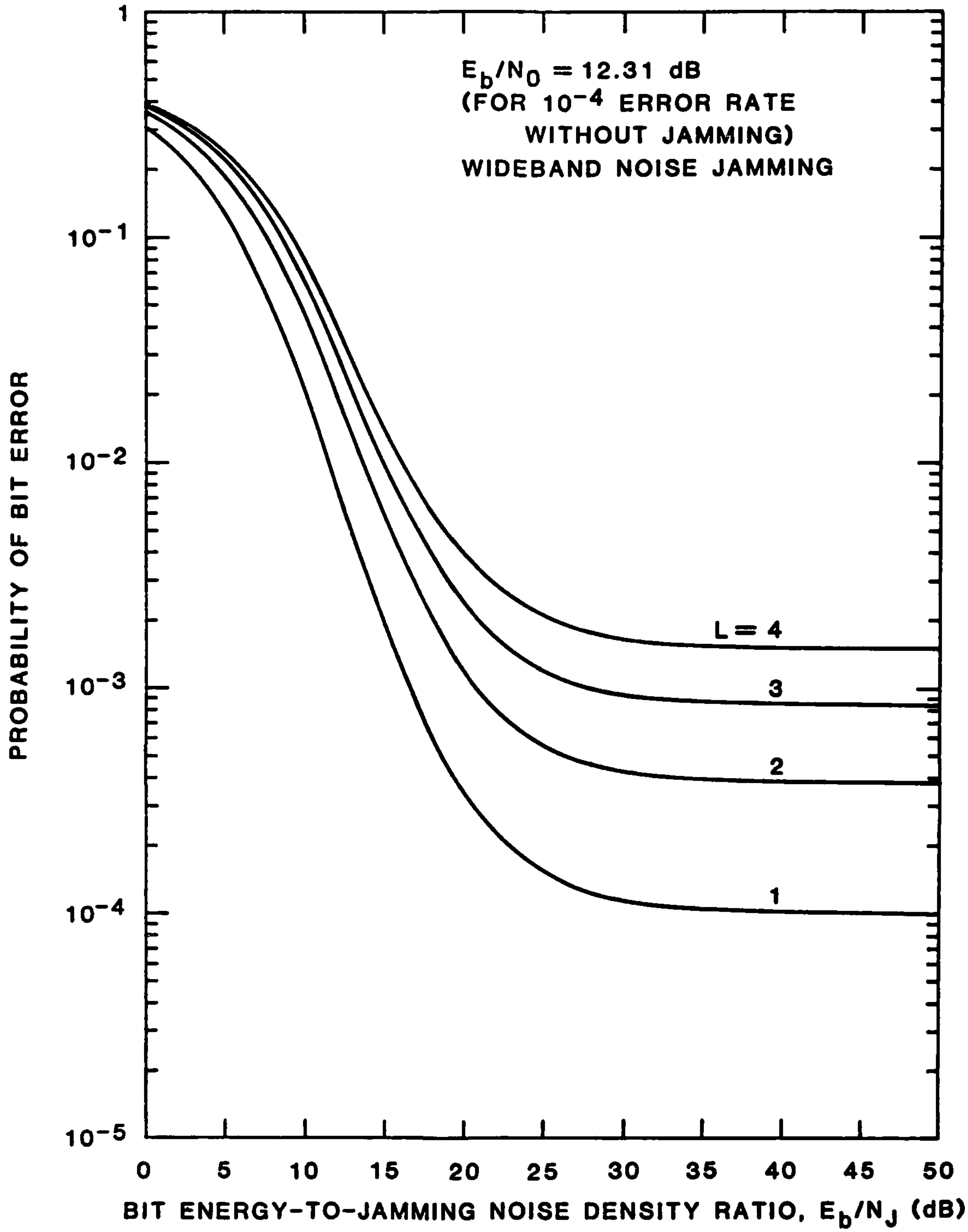


FIGURE 5-14 WIDEBAND NOISE JAMMING PERFORMANCE OF FH/BFSK SELF-NORMALIZING RECEIVER WHEN $E_b/N_0 = 12.31$ dB WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER

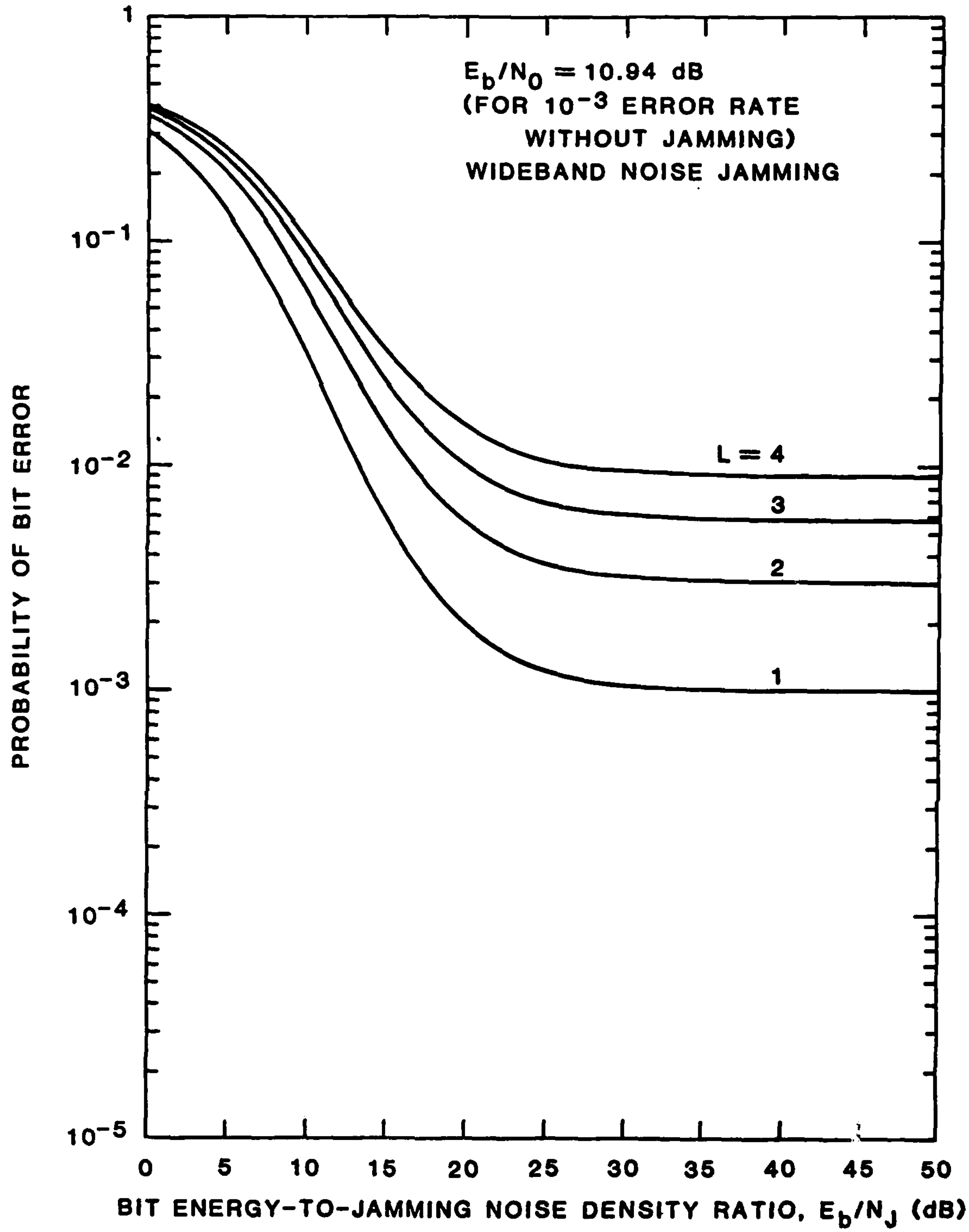


FIGURE 5-15 WIDEBAND NOISE JAMMING PERFORMANCE OF FH/BFSK SELF-NORMALIZING RECEIVER WHEN $E_b/N_0 = 10.94$ dB WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER

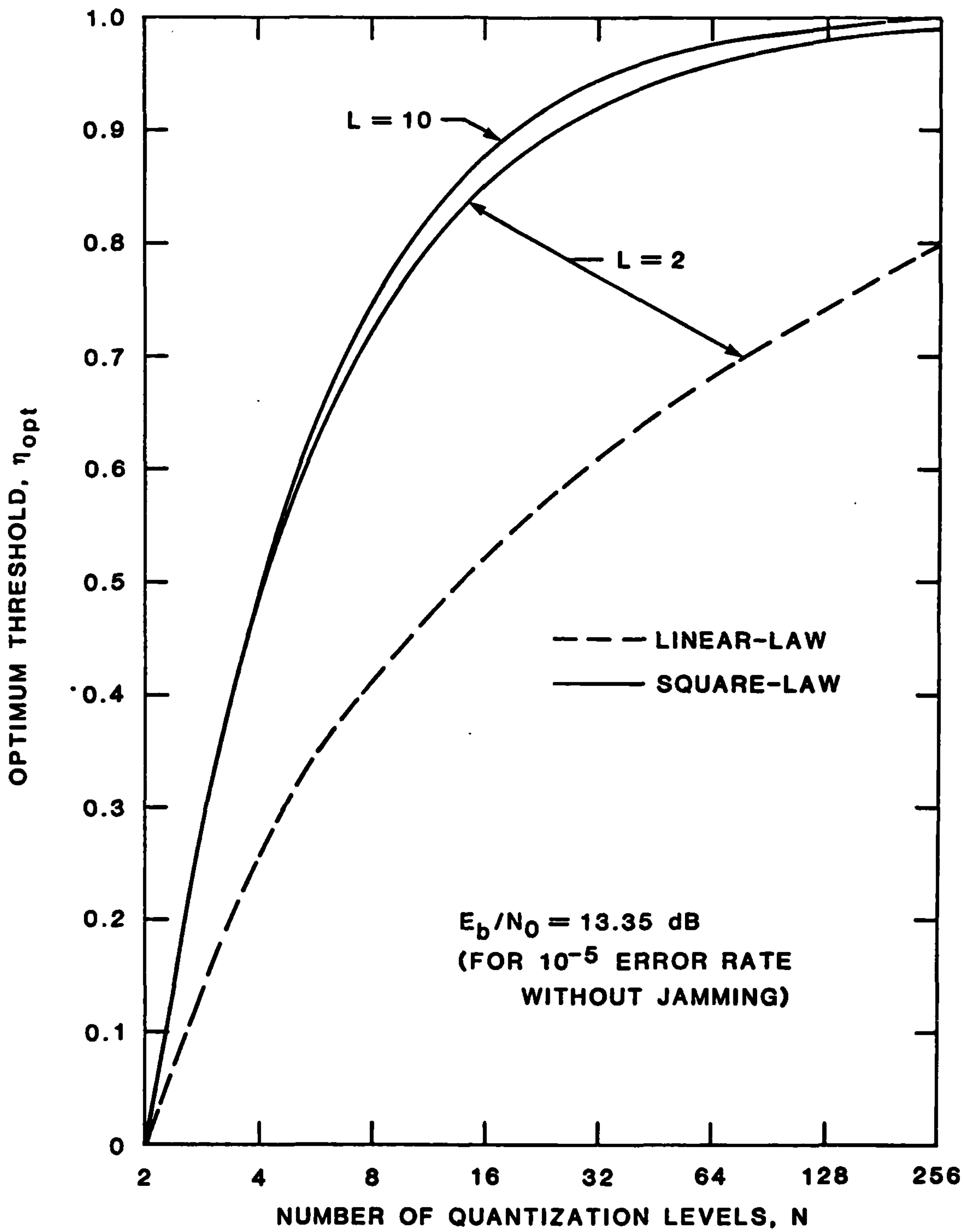


FIGURE 5-16 OPTIMUM THRESHOLD VS. NUMBER OF QUANTIZATION LEVELS FOR FH/BFSK SELF-NORMALIZING RECEIVER WHEN $E_b/N_0 = 13.35$ dB WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER

This plot suggests that the linear-law receiver requires a higher number of quantization levels than the square-law receiver for the same threshold.

Figures 5-17 through 5-19 show a comparison of wideband jamming performances of the quantized and unquantized versions of the square-law FH/BFSK self-normalizing receiver for L=2, 4, and 6 with N as a parameter. It is shwon that as N increases, the difference becomes smaller and for 32 levels of quantization, the difference becomes negligible; thus, it is adequate to use an N=32 quantized receiver model to approximate the unquantized performance.

Figures 5-20 through 5-22 show the same comparison for optimum jamming performance. From these figures, N=32 is also adequate to achieve negligible degradation for the square-law receiver with a quantizer for digital implementation.

We have shown in earlier subsections that for L greater than 4, the numerical computation for the unquantized self-normalizing FH/BFSK receiver employing a square-law detector becomes quite tedious and computation for L beyond 4 was therefore abandoned. The quantized version, however, posed no difficulties for any given L, and, thus, served as a computation tool since the performance of the quantized version is an upper bound to the unquantized performance. A program listing is provided in Appendix 5C so that the user may run it for different values of L not given in this report or for variations of other parameters such as E_b/N_0, E_b/N_J, N, η_{opt}, etc.

Figures 5-23 through 5-26 show, for fixed values of L, the quantized performance as a function of E_b/N_J with γ as parameter. These curves indicate that for L=1, wideband jamming (γ=1) is optimum only for E_b/N_J less than 5 dB and as L is increased, the range of E_b/N_J for which γ_0=1 becomes wider and thus the range of usefulness of partial-band jamming diminishes.

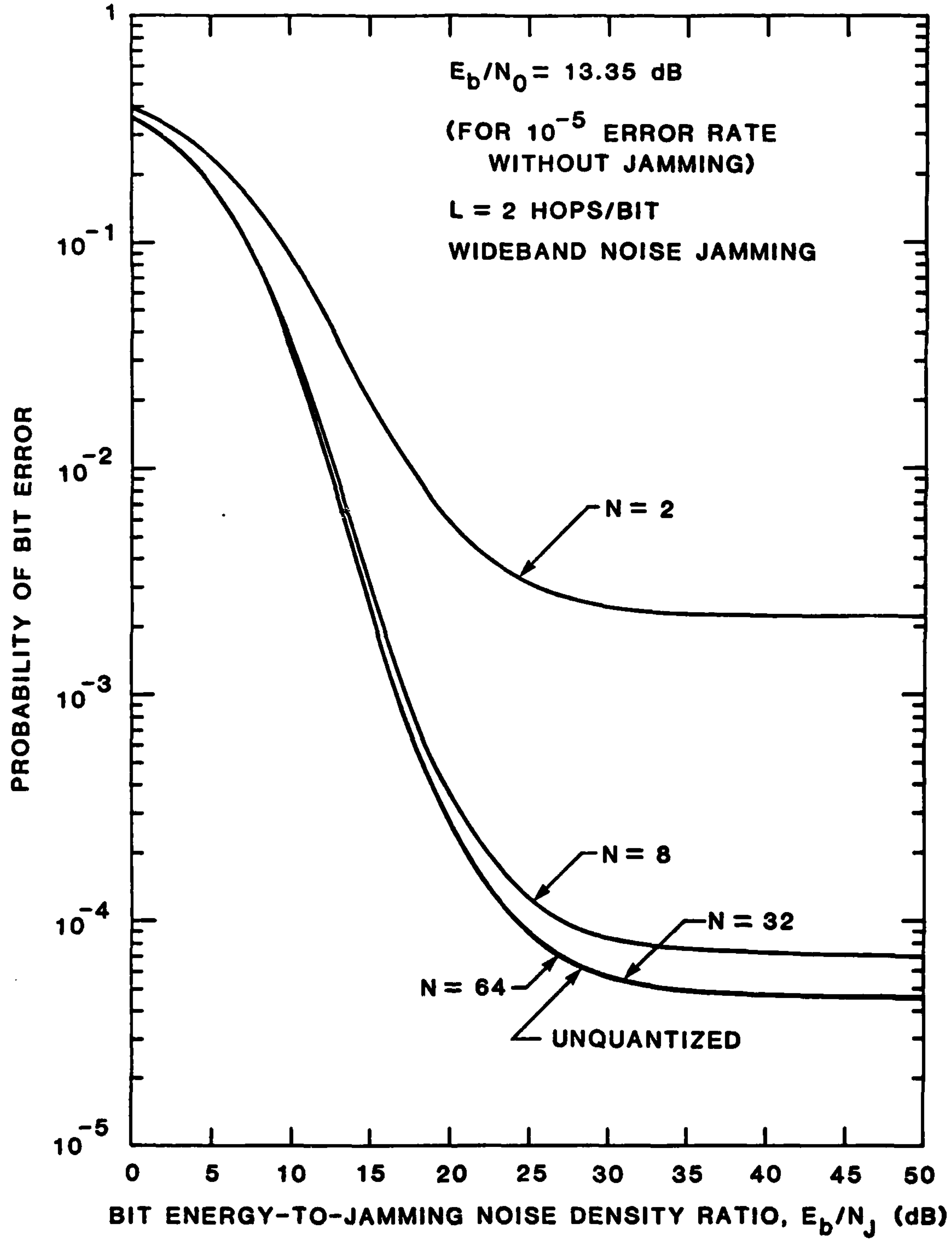


FIGURE 5-17 WIDEBAND NOISE JAMMING PERFORMANCE OF FH/BFSK SQUARE-LAW SELF-NORMALIZING RECEIVER WITH N-LEVEL QUANTIZER FOR L = 2 HOPS/BIT WHEN E_b/N_0 = 13.35 dB

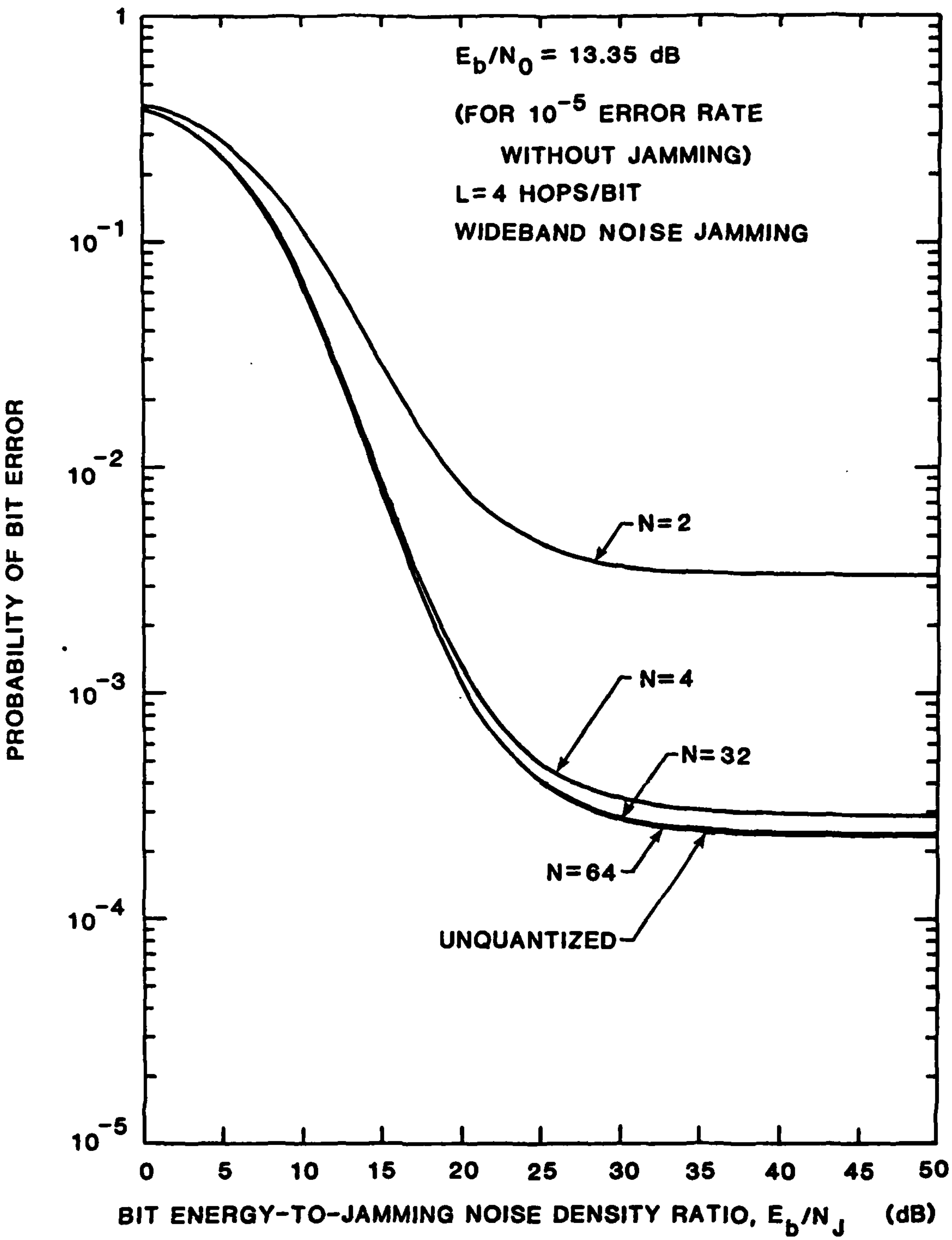


FIGURE 5-18 WIDEBAND NOISE JAMMING PERFORMANCE OF FH/BFSK SQUARE-LAW SELF-NORMALIZING RECEIVER WITH N-LEVEL QUANTIZER FOR L=4 HOPS/BIT WHEN $E_b/N_0 = 13.35$ dB

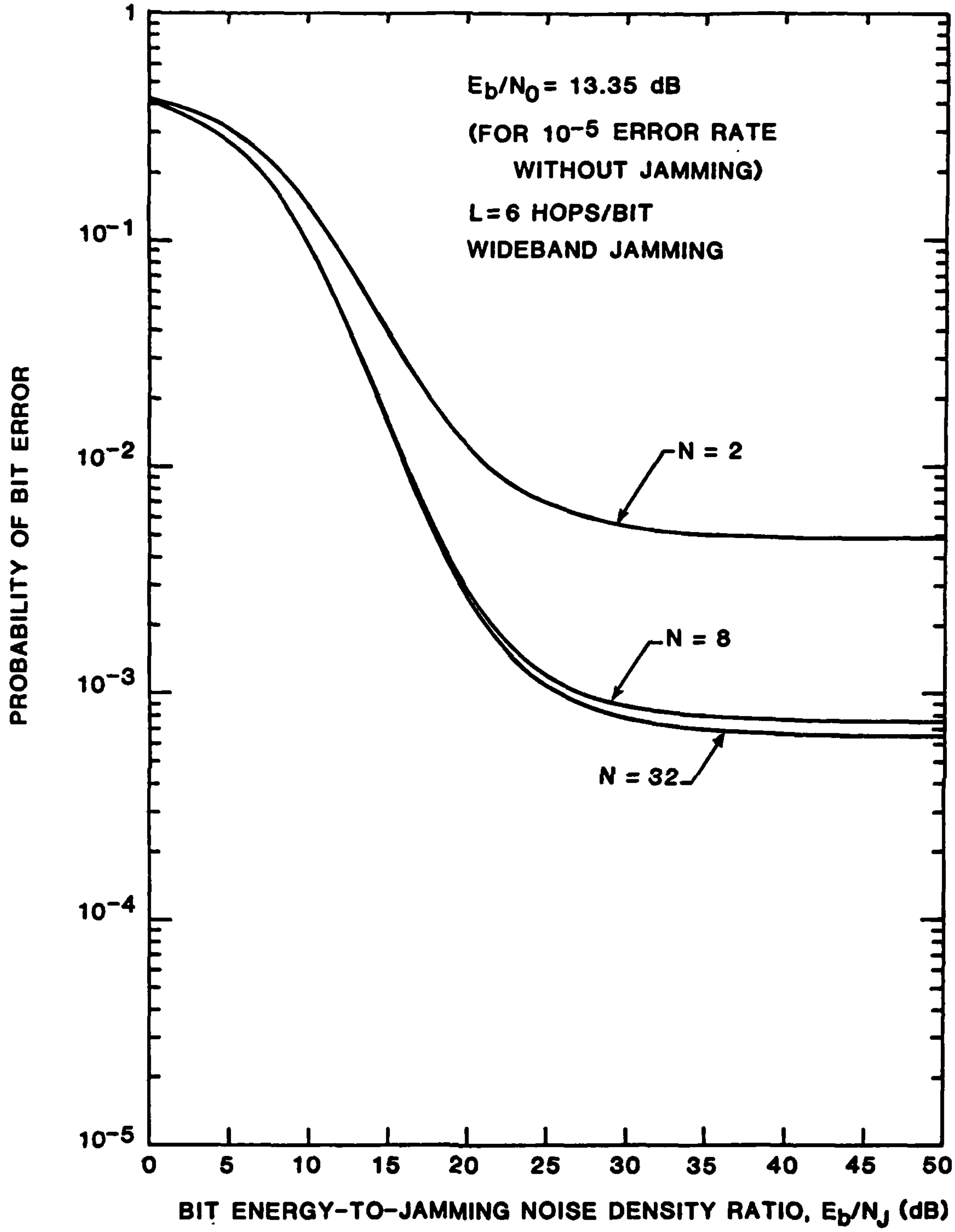


FIGURE 5-19 WIDEBAND NOISE JAMMING PERFORMANCE OF FH/BFSK SQUARE-LAW SELF-NORMALIZING RECEIVER WITH N-LEVEL QUANTIZER FOR L=6 HOPS/BIT WHEN E_b/N_0 = 13.35 dB

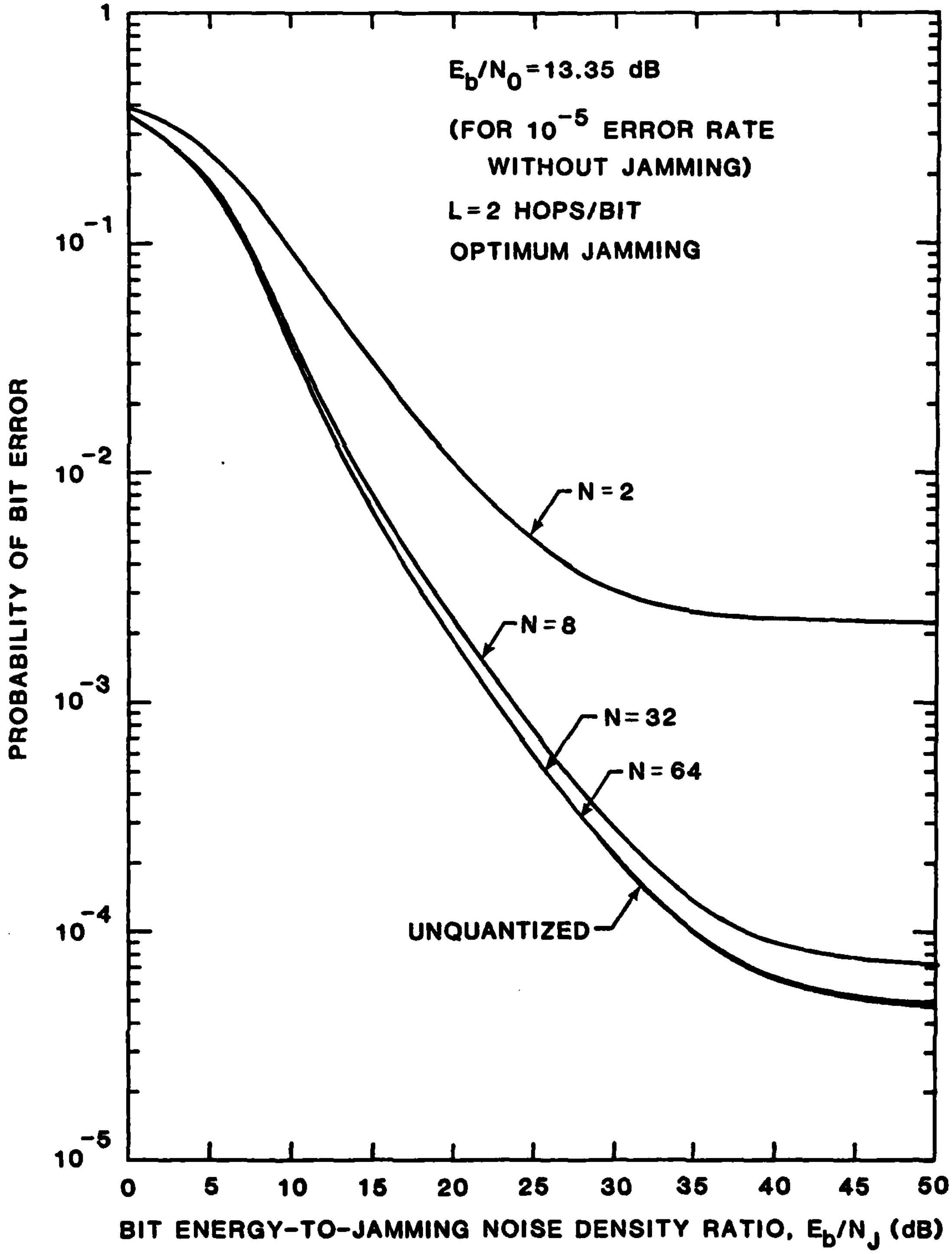


FIGURE 5-20 OPTIMUM JAMMING PERFORMANCE OF FH/BFSK SQUARE-LAW SELF-NORMALIZING RECEIVER WITH N-LEVEL QUANTIZER FOR L = 2 HOPS/BIT WHEN $E_b/N_0 = 13.35$ dB WITH N AS A PARAMETER

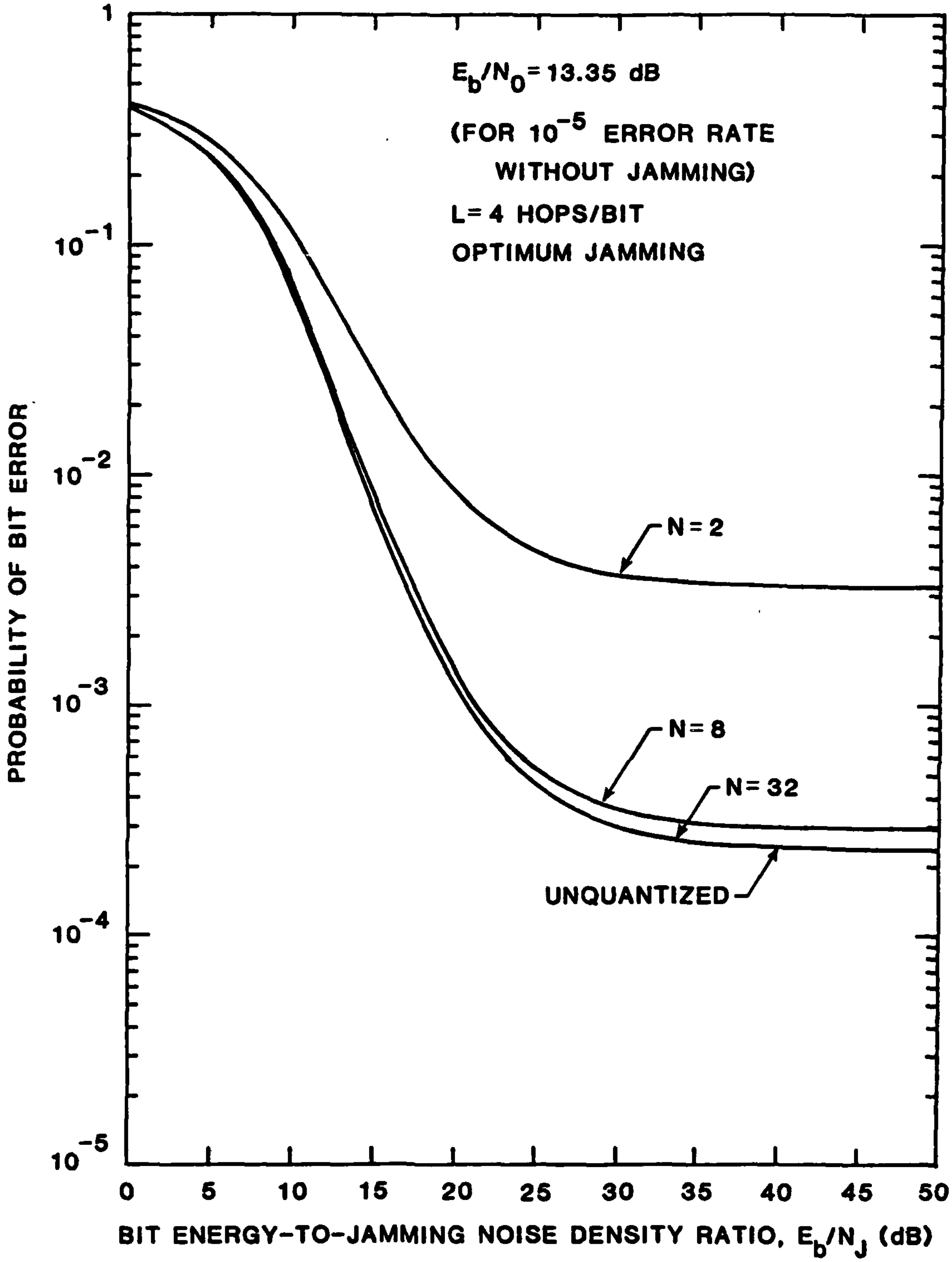


FIGURE 5-21 OPTIMUM JAMMING PERFORMANCE OF FH/BFSK SQUARE-LAW SELF-NORMALIZING RECEIVER WITH N-LEVEL QUANTIZER FOR L=4 HOPS/BIT WHEN E_b/N_0= 13.35 dB WITH N AS A PARAMETER

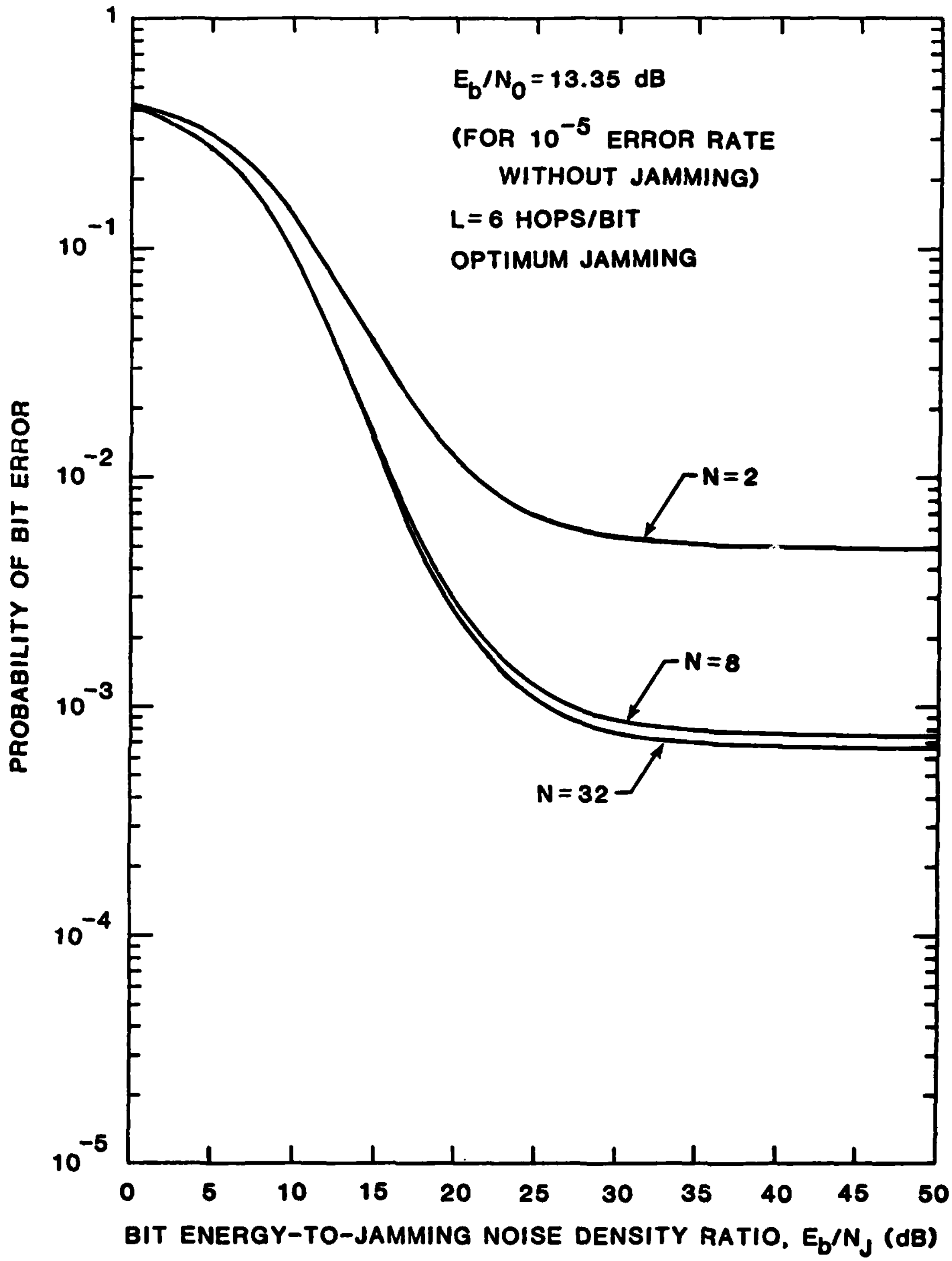


FIGURE 5-22 OPTIMUM JAMMING PERFORMANCE OF FH/BFSK SQUARE-LAW SELF-NORMALIZING RECEIVER WITH N-LEVEL QUANTIZER FOR L=6 HOPS/BIT WHEN E_b/N_0= 13.35 dB WITH N AS A PARAMETER

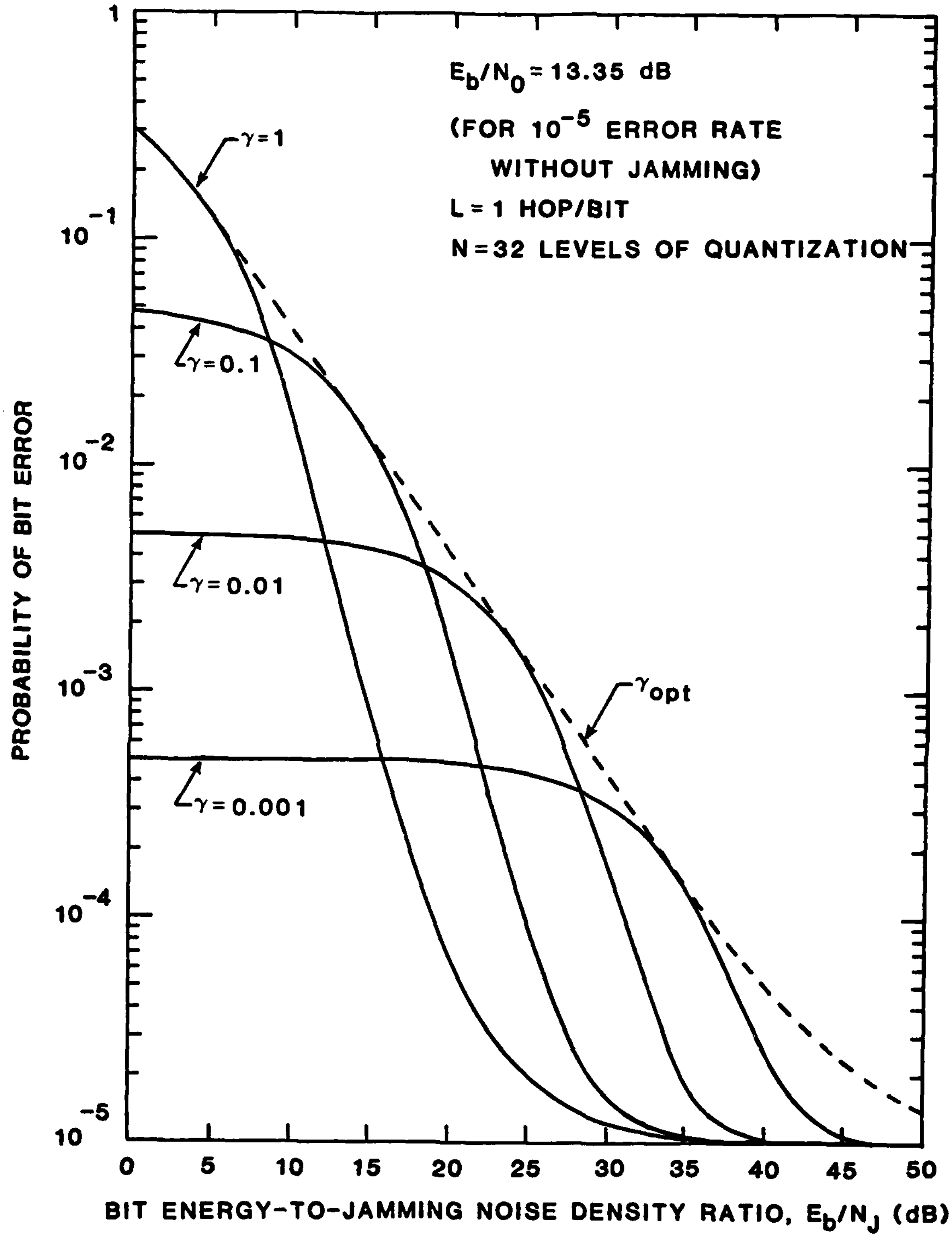


FIGURE 5-23 PROBABILITY OF BIT ERROR VS. E_b/N_J FOR L = 1 HOP/BIT AND 32-LEVEL QUANTIZER FOR FH/BFSK SQUARE-LAW SELF-NORMALIZING RECEIVER WITH γ AS A PARAMETER

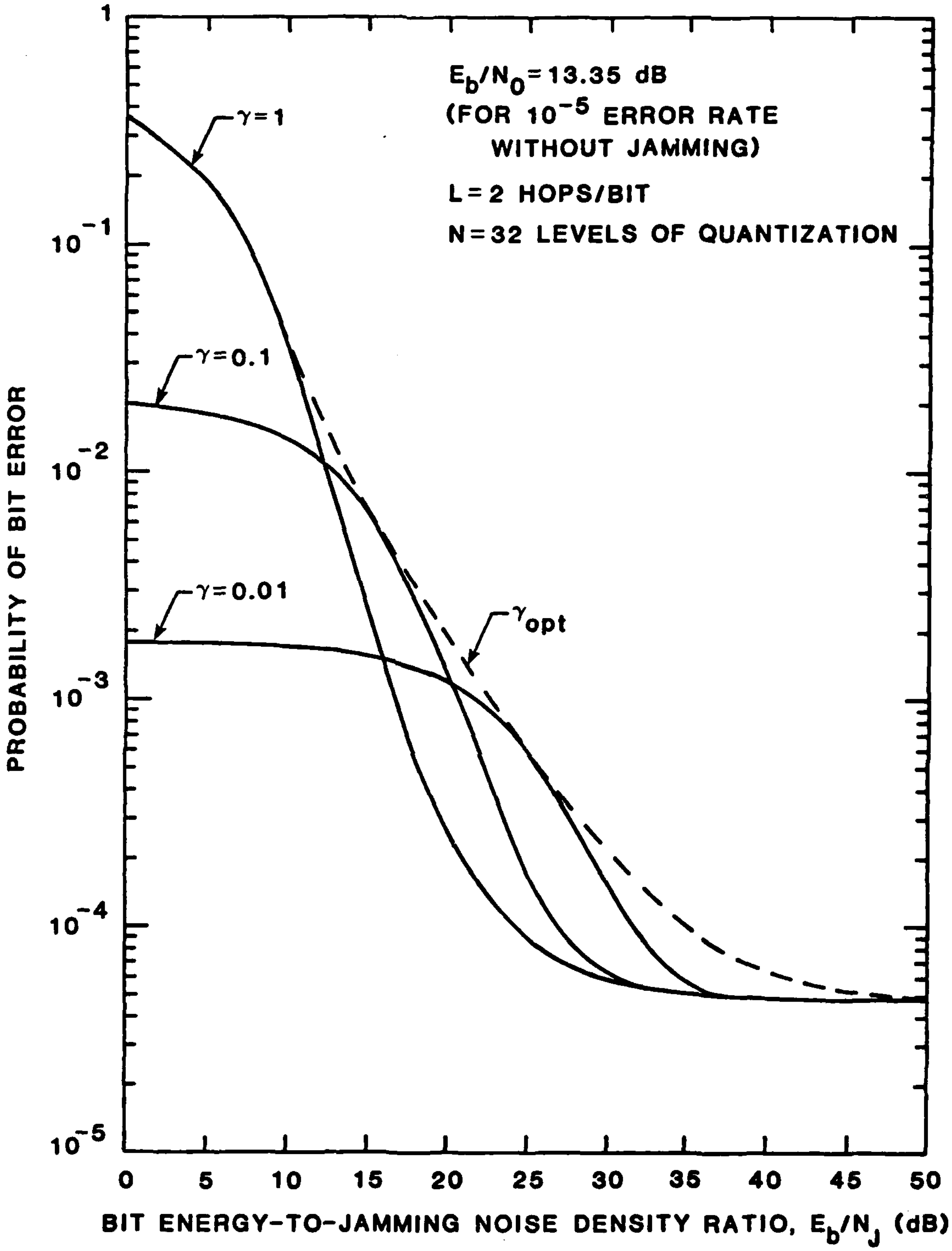


FIGURE 5-24 PROBABILITY OF BIT ERROR VS. E_b/N_J FOR L = 2 HOPS/BIT AND 32-LEVEL QUANTIZER FOR FH/BFSK SQUARE-LAW SELF-NORMALIZING RECEIVER WITH γ AS A PARAMETER

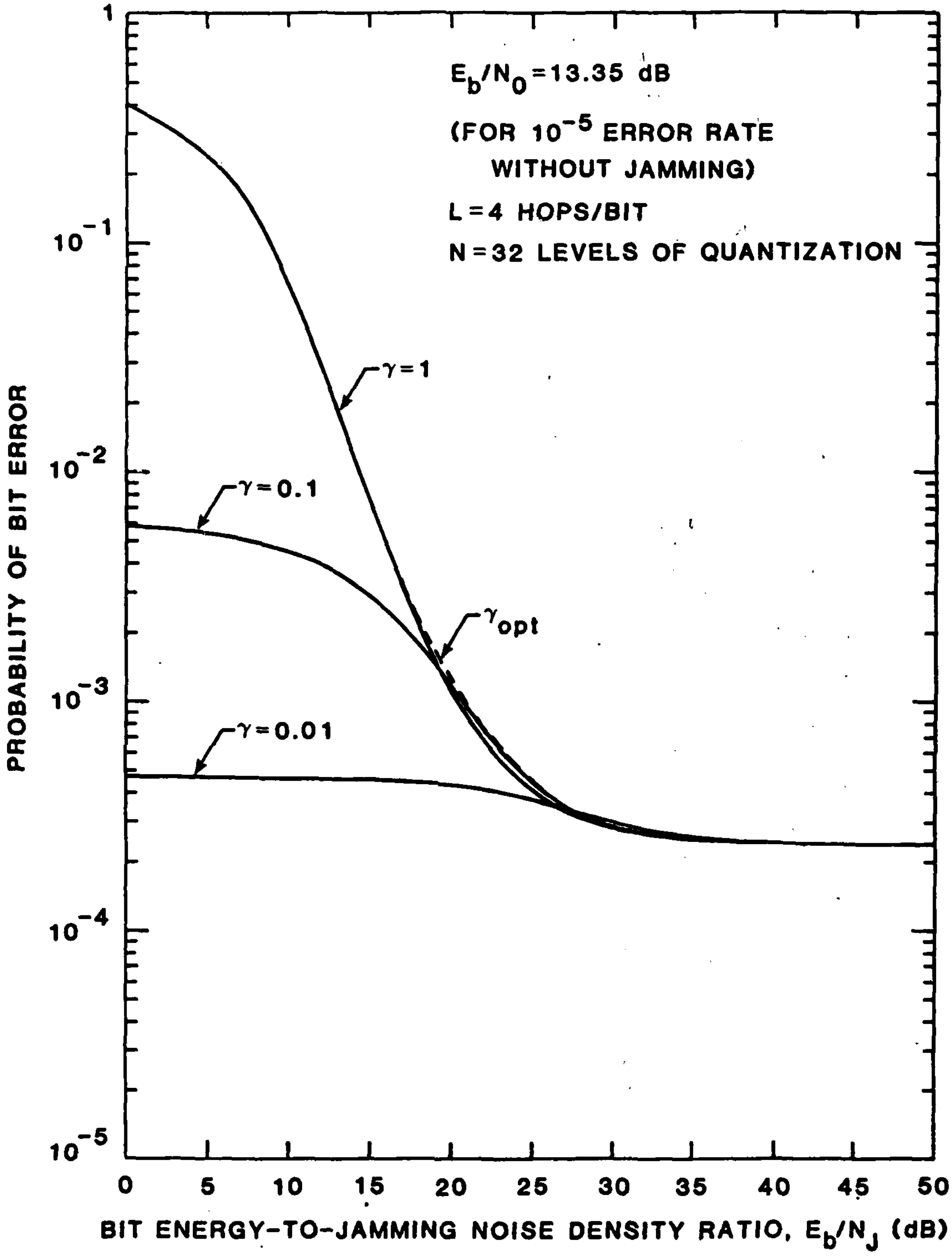


FIGURE 5-25 PROBABILITY OF BIT ERROR VS. E_b/N_J FOR L = 4 HOPS/BIT AND 32-LEVEL QUANTIZER FOR FH/BFSK SQUARE-LAW SELF-NORMALIZING RECEIVER WITH γ AS A PARAMETER

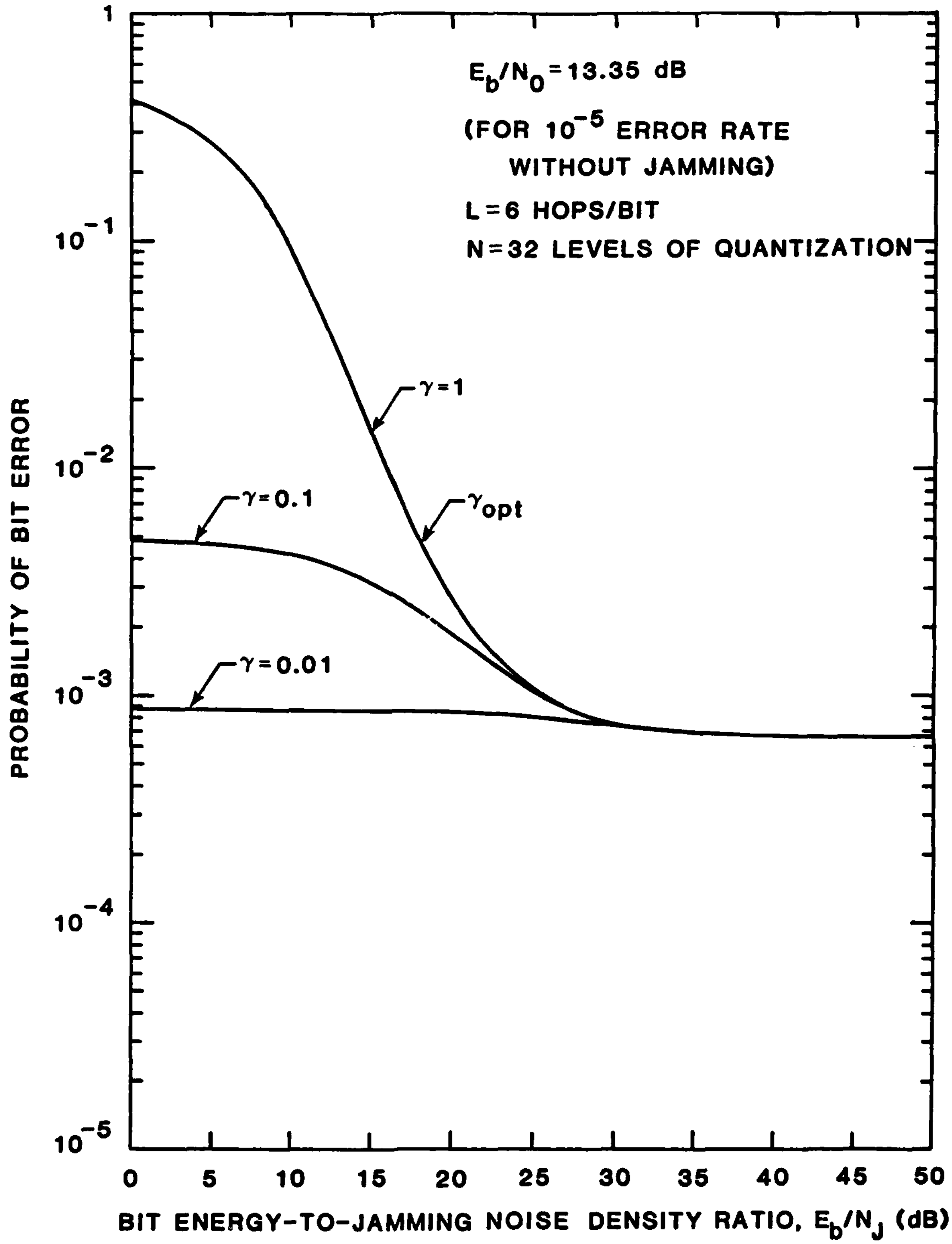


FIGURE 5-26 PROBABILITY OF BIT ERROR VS. E_b/N_J FOR L = 6 HOPS/BIT AND 32-LEVEL QUANTIZER FOR FH/BFSK SQUARE-LAW SELF-NORMALIZING RECEIVER WITH γ AS A PARAMETER

5.4.3 Comparison of Linear-Law and Square-Law Receivers

It is well known that for the binary case there is no performance difference between the linear-law receiver and the square-law receiver when L=1. It is also true that there is no difference in performance when L=2. This can be shown as follows. The bit error probability for the linear-law case is

$$P_b(e) = \Pr\left\{\frac{x_{11} - x_{21}}{x_{11} + x_{21}} + \frac{x_{12} - x_{22}}{x_{12} + x_{22}} < 0\right\}$$

$$= \Pr\left\{x_{11}x_{12} < x_{21}x_{22}\right\} ; \qquad (5\text{-}73)$$

and, for square-law case,

$$P_b(e) = \Pr\left\{\frac{x_{11}^2 - x_{21}^2}{x_{11}^2 + x_{21}^2} + \frac{x_{12}^2 - x_{22}^2}{x_{12}^2 + x_{22}^2} < 0\right\}$$

$$= \Pr\left\{x_{11}x_{12} < x_{21}x_{22}\right\}. \qquad (5\text{-}74)$$

Both of these lead to the same error event, namely $x_{11}x_{12} < x_{21}x_{22}$. This result has been confirmed numerically without using a quantizer. When a discrete-level quantizer with N levels is included, both receivers approach the same asymptote. It is interesting to note that the square-law receiver approaches the asymptote at N=256.

When L is greater than 2, the exact analysis without quantizer for the linear-law receiver is very involved. The linear-law results, then, rely heavily on the performance of the linear-law receiver with quantizer. As an example comparison between linear-law and square-law self-normalizing receivers, Table 5-1 shows that the linear-law receiver utilizing a 256-level quantizer performs slightly worse than the square-law receiver employing a 64-level quantizer for L=4. However, it can be shown that the optimum threshold for the

TABLE 5-1

BFSK/FH LINEAR-LAW RECEIVER WITH SELF-NORMALIZATION AND 256-LEVEL QUANTIZER VS. BFSK/FH SQUARE-LAW RECEIVER WITH SELF-NORMALIZATION AND 64-LEVEL QUANTIZER FOR 4 HOPS/BIT WITH E_b/N_0 AS PARAMETER

E_b/N_0 (dB)	LINEAR-LAW WITH SELF-NORMALIZATION (N=256 LEVELS)		SQUARE-LAW WITH SELF-NORMALIZATION (N = 64 LEVELS)	
	OPTIMUM THRESHOLD	BER	OPTIMUM THRESHOLD	BER
-5.0	0.514	0.464×10^{0}	0.965	0.464×10^{0}
0.0	0.514	0.391×10^{0}	0.965	0.391×10^{0}
5.0	0.517	0.215×10^{0}	0.968	0.215×10^{0}
10.0	0.526	0.222×10^{-1}	0.974	0.221×10^{-1}
15.0	0.530	0.362×10^{-5}	0.977	0.350×10^{-5}

linear-law receiver has not reached its final value yet, whereas the square-law receiver has almost reached its maximum. This means that the quantization error is still more significant for the linear-law receiver than the square-law receiver under these conditions.

6.0 COMPARISON OF ECCM RECEIVER PROCESSING SCHEMES

In the previous sections, we considered the exact performance analyses and numerical results for four different types of FH/MFSK receiver: the conventional square-law combining receiver and three nonlinear combining receivers, including per-hop processing with a clipper (soft-limiter), with AGC (Adaptive Gain Control), and with a self-normalizing scheme. The distinction between linear and nonlinear combining is based on the manner in which the weighted sum is obtained for the decision statistics. Figure 6-1 shows a generic FH/MFSK receiver model and Table 6-1 describes these four different types of receivers. The clipper receiver modifies the standard FH/MFSK receiver by inserting clippers prior to accumulating the square-law envelope detector outputs. In the AGC receiver, the detector outputs are normalized by ideal measurements of the received noise power on a per-hop basis. The self-normalizing receiver uses the sum of the detector outputs for normalization on each hop. These nonlinear combining techniques are designed to suppress the jamming effects to enhance the receiver performance.

Our analyses of these FH/MFSK systems in the partial-band noise jamming channel include the presence of the system's thermal noise and are based on direct calculation for the error performances. This is in contrast to previous works, e.g. [9], [10], [11], which were mainly carried out by assuming that the system's thermal noise was absent, and using a bounding technique (i.e., union bound) to obtain approximate results for $M > 2$. The specific interest in the previous works was to obtain the bit error probability produced by the conventional square-law linear combining receiver under worst-case partial-band jamming and to determine the role of the diversity (L) in combatting the jamming effects. (Against fading, the use of L hops per symbol was known to be an optimum method for improving the performance.)

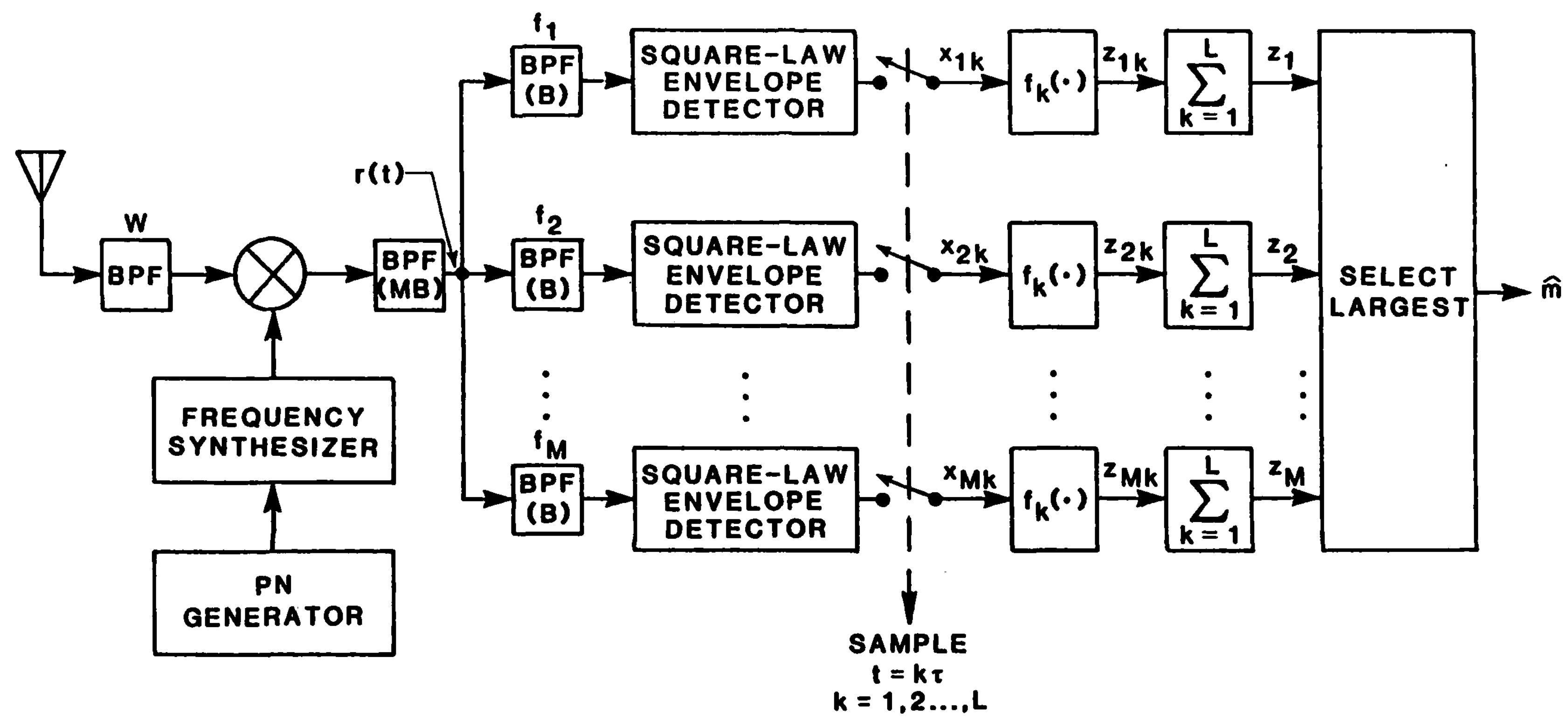


FIGURE 6-1 FH/MFSK SQUARE-LAW RECEIVERS CONSIDERED

TABLE 6-1

DESCRIPTIONS OF THE RECEIVERS

RECEIVER TYPE	SPECIFICATION OF $z_{ik} = f_k(x_{ik}),\ i=1,2,\ldots,M$	REMARKS	IS SIDE INFORMATION ON JAMMING STATE USED IN DECISION?
LINEAR COMBINING RECEIVER	$z_{ik} = x_{ik}$	Direct Connection (Linear Combining)	No
CLIPPER RECEIVER	$z_{ik} = \begin{cases} x_{ik}, & x_{ik} \le n \\ n, & x_{ik} > n \end{cases}$	Soft Limiter (Nonlinear Combining)	No
AGC RECEIVER	$z_{ik} = x_{ik}/\sigma_k^2$ $\sigma_k^2 = \begin{cases} \sigma_N^2, & \text{if not jammed} \\ \sigma_N^2 + \sigma_J^2, & \text{if jammed} \end{cases}$ (σ_k^2 = measured)	Adaptive Gain Control (Nonlinear Combining)	No
SELF-NORMALIZING RECEIVER	$z_{ik} = \dfrac{x_{ik}}{\sum_{i=1}^{M} x_{ik}}$	Practical Realization of AGC Using In-Band Measurements	No

We may ask, "Can we obtain the performance measure of FH/MFSK receivers by using a union bound?" For a set of M orthogonal waveforms, applying the union bound to the bit error probability yields

$$P_b(e;M) = \frac{M}{2(M-1)} P_s(e)$$

$$\leq \frac{M}{2(M-1)} (M-1)\ P_2(e) = \frac{M}{2} P_2(e). \tag{6-1}$$

Here $P_b(e)$ denotes bit error probability, $P_s(e)$ denotes symbol error probability and $P_2(e)$ denotes the error probability for a binary system. Figure 6-2, which is based on the FH/BFSK linear combining receiver, shows the bit error probability performance of FH/MFSK using the union bound. It was shown in the previous sections that for the different receiver types the exact analytical results gave better error performance with increasing alphabet size, M. But in Figure 6-2, the union bound to this FH/MFSK signaling is shown to give the erroneous conclusion that performance degrades with increasing M.

A recent paper by Crepeau and McGregor [12] discloses that application of the union bound to MFSK signaling on certain channels (worst-case partial-band Gaussian jamming channel or the Rayleigh fading channel) where the probability of error varies in an inverse linear fashion with E_b/N_0 can lead to the erroneous conclusion that performance degrades with increasing M. The error performance curves we observed in our FH/MFSK worst-case partial-band jamming analyses were linear. So the conclusion is that only exact analysis is sufficient to evaluate FH/MFSK system performance.

Our previous exact analytical results for FH/BFSK systems including the effects of thermal noise [1] show that the square-law linear combining receiver is the least effective, as compared to two nonlinear combining receivers

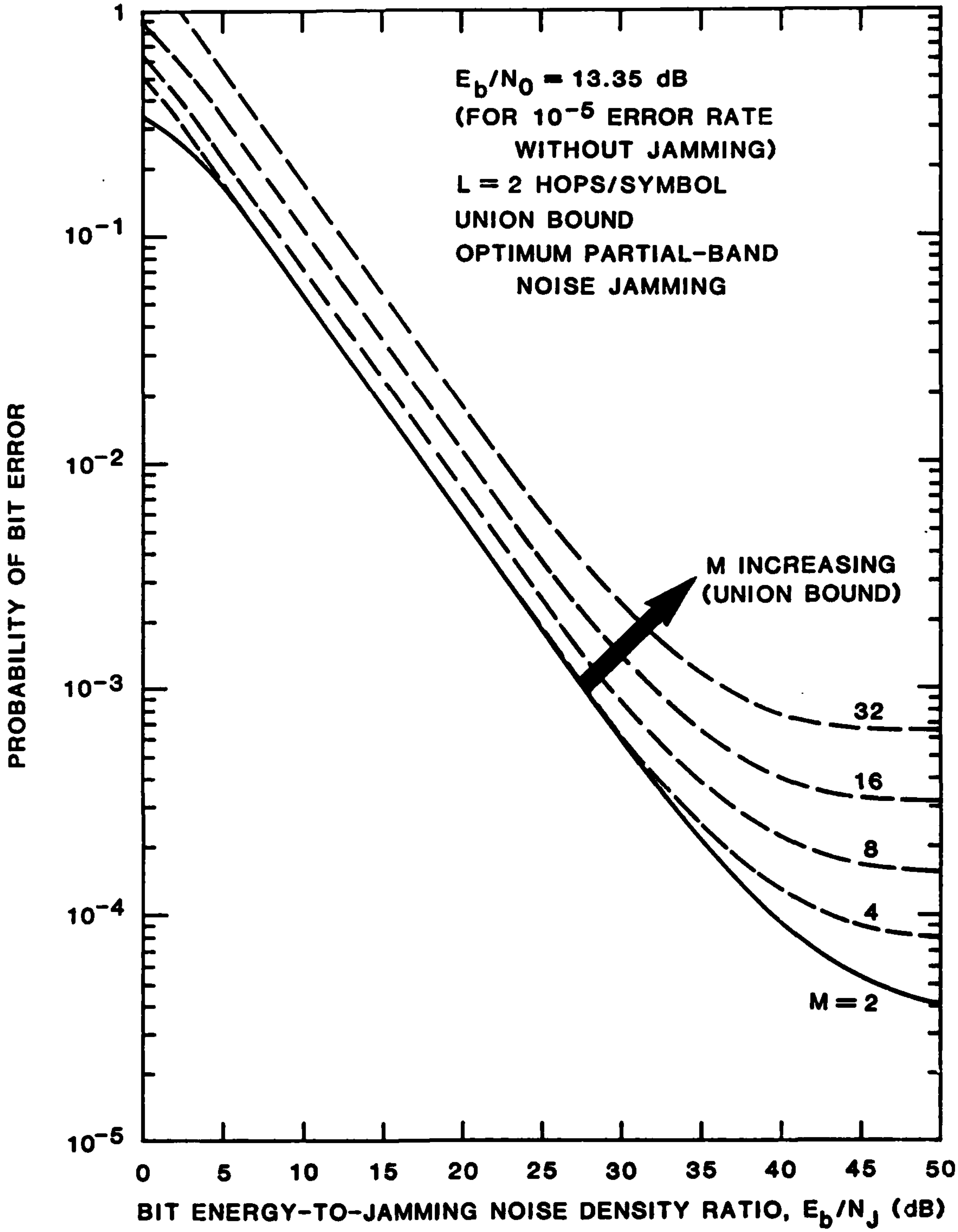


FIGURE 6-2 BIT ERROR PROBABILITY USING UNION BOUND FOR SQUARE-LAW LINEAR COMBINING RECEIVER

(clipper receiver and AGC receiver), in demodulating the postulated waveform in the worst-case partial-band noise jamming.

In this section, we compare the performances of three different types of receiver (linear combining receiver, AGC receiver and clipper receiver) for FH/MFSK signals in the worst-case (jammer's optimum) partial-band noise jamming environment. These performance comparisons are shown for both linear-law and square-law detectors.

6.1 COMPARISONS FOR SQUARE-LAW COMBINING RECEIVERS

Due to complexities in numerical computations of bit error probability for the conventional linear combining receiver even with square-law detectors, we compare the performances of only the square-law combining AGC receiver and the square-law combining clipper receiver when $L > 2$.

6.1.1 Optimum Jamming Fraction

Figures 6-3 and 6-4 show the typical behavior of the optimum partial-band jamming fraction, γ_0, as a function of L, the number of hops/symbol, for M=4 and M=8, respectively. It is seen that, in general, for both the AGC receiver and the clipper receiver, the value of γ_0 increases as L increases for a given E_b/N_J. But for the AGC receiver the value of γ_0 becomes equal to one for small L. Thus generally the AGC receiver is vulnerable to wideband jamming, while the clipper receiver is vulnerable to partial-band jamming.

6.1.2 Error Probability

Figures 6-5 through 6-7 show the performances of three different square-law receivers (linear combining (Figures 6-5 and 6-6 only), clipper, and AGC receivers) under worst-case partial-band noise jamming with M=8 as a typical alphabet size for L=1, 2, and 4 hops/symbol, respectively. In Figure 6-5

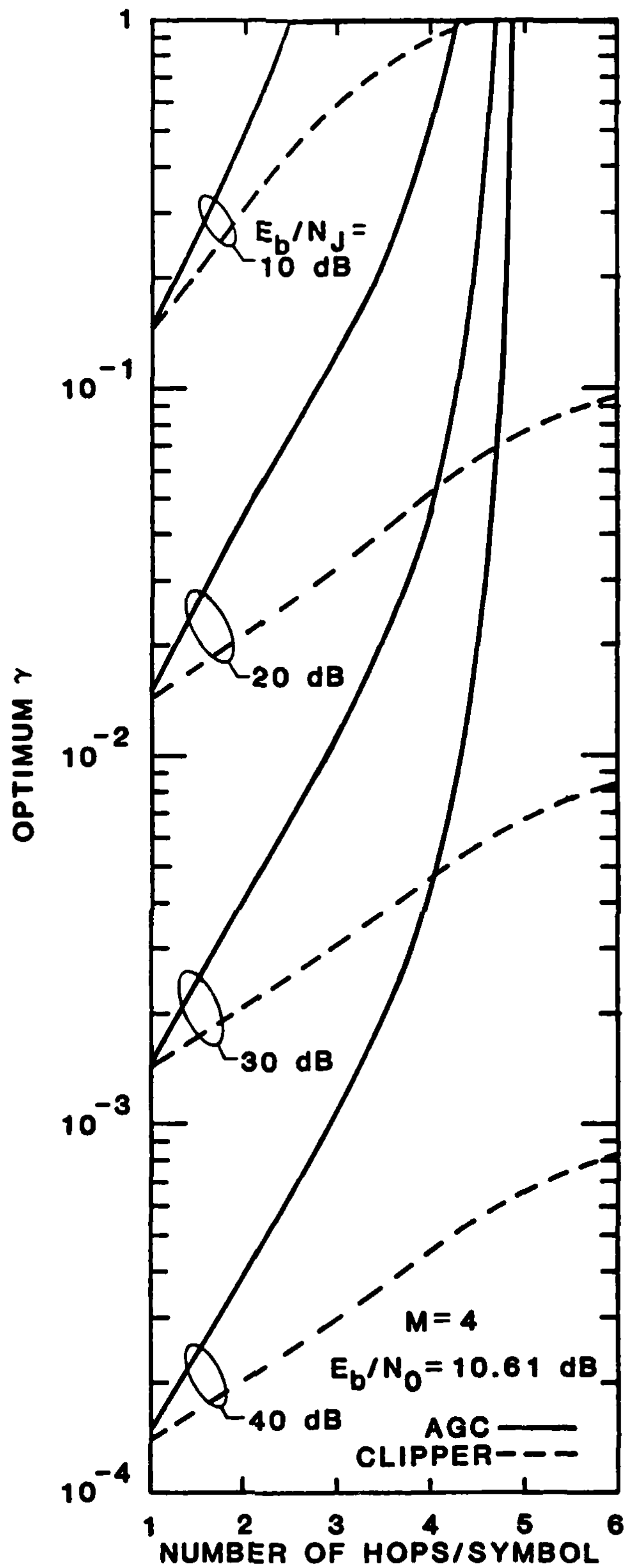


FIGURE 6-3 OPTIMUM JAMMING FRACTION (γ) VS. NUMBER OF HOPS/SYMBOL (L) FOR THE AGC AND CLIPPER FH/MFSK (M = 4) RECEIVERS WHEN E_b/N_0 = 10.61 dB WITH E_b/N_J AS A PARAMETER (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

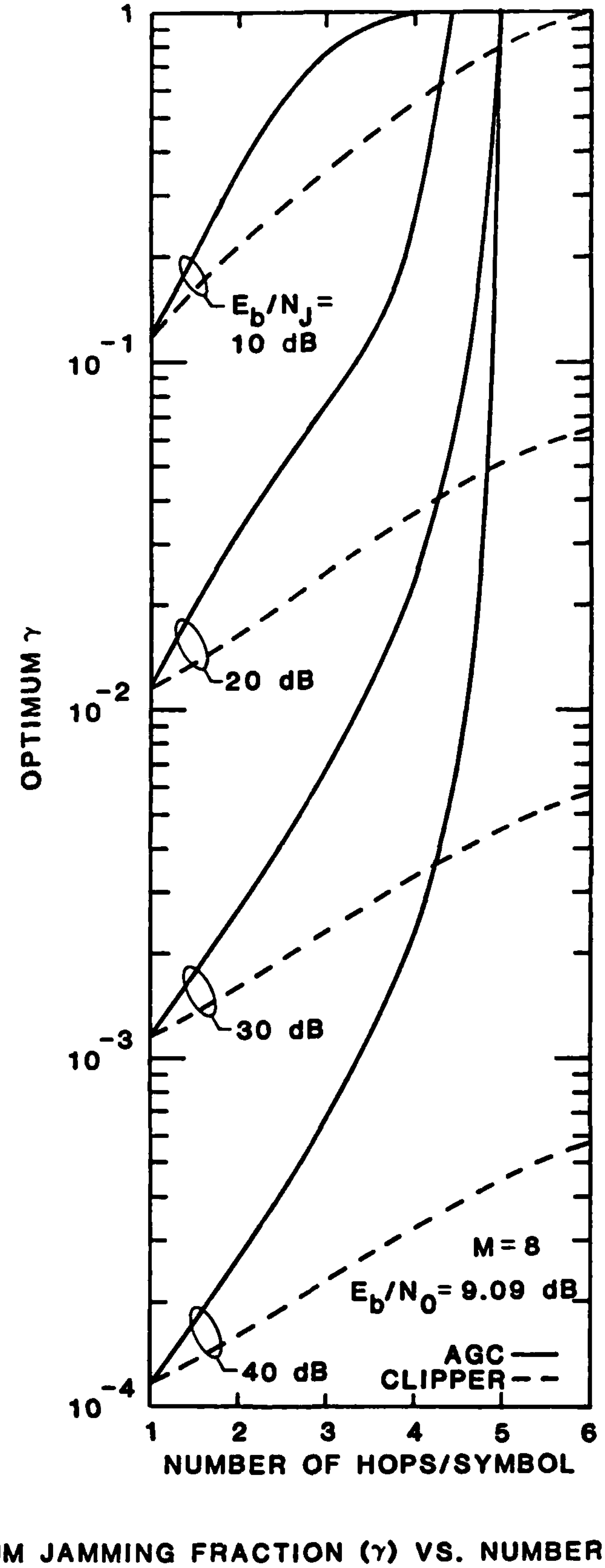


FIGURE 6-4 OPTIMUM JAMMING FRACTION (γ) VS. NUMBER OF HOPS/SYMBOL (L) FOR THE AGC AND CLIPPER FH/MFSK (M=8) RECEIVERS WHEN E_b/N_0=9.09 dB WITH E_b/N_J AS A PARAMETER (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

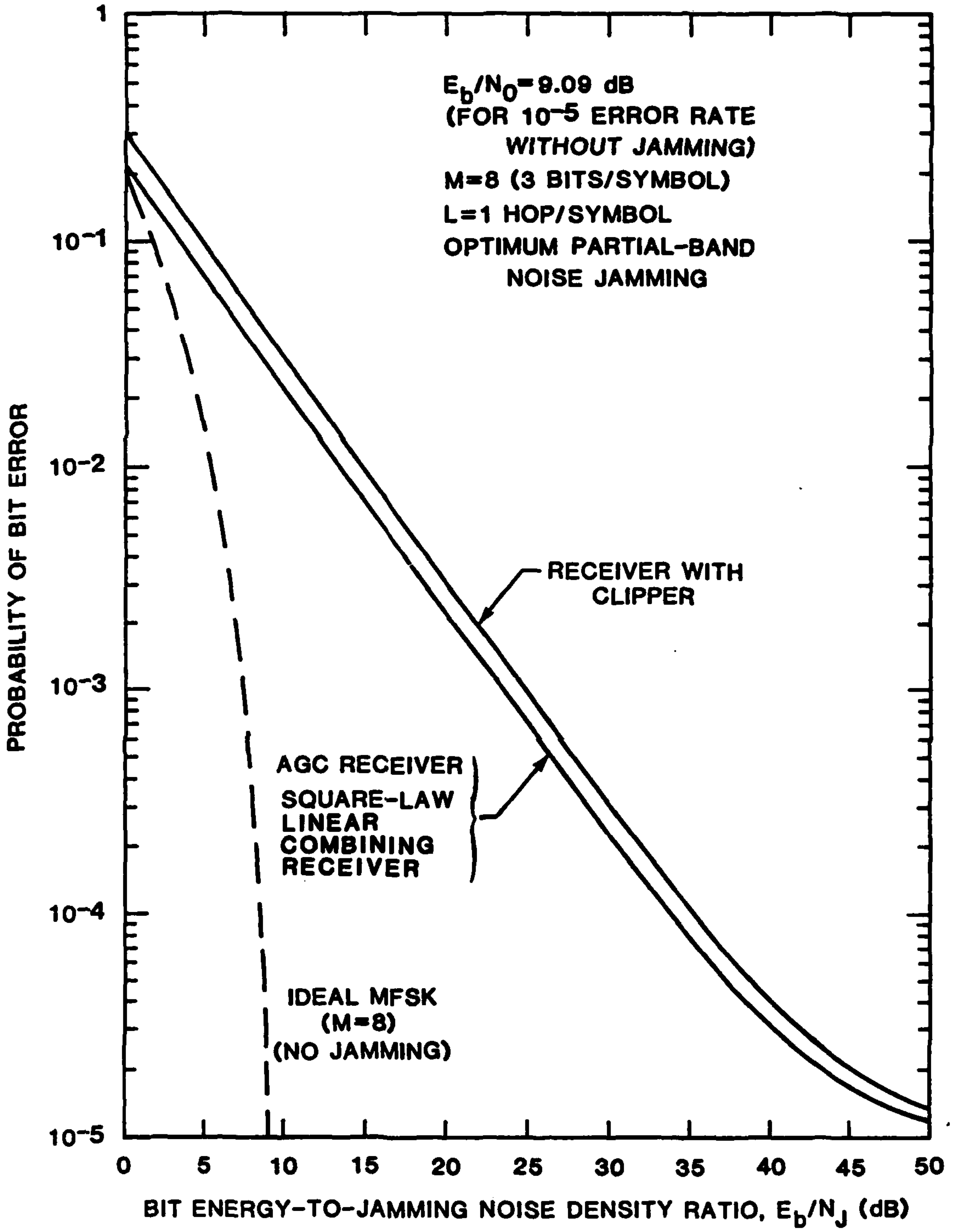


FIGURE 6-5 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/MFSK (M=8) SQUARE-LAW COMBINING RECEIVERS FOR L = 1 HOP/SYMBOL WHEN E_b/N_0= 9.09 dB (FOR IDEAL MFSK (M=8) CURVE THE ABSCISSA READS E_b/N_0)

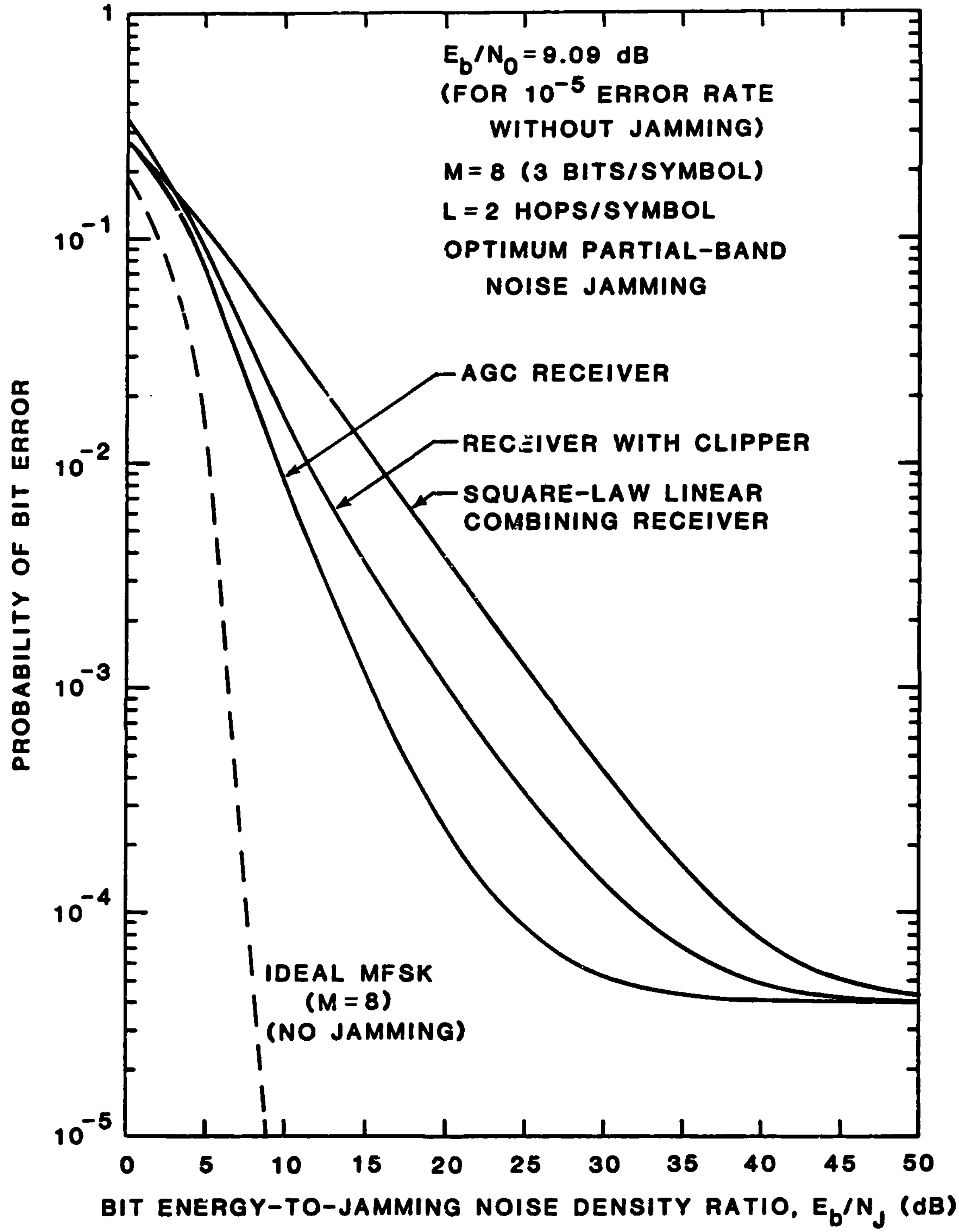


FIGURE 6-6 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/MFSK (M= 8) SQUARE-LAW COMBINING RECEIVERS FOR L= 2 HOPS/SYMBOL WHEN E_b/N_0= 9.09 dB (FOR IDEAL MFSK (M= 8) CURVE THE ABSCISSA READS E_b/N_0)

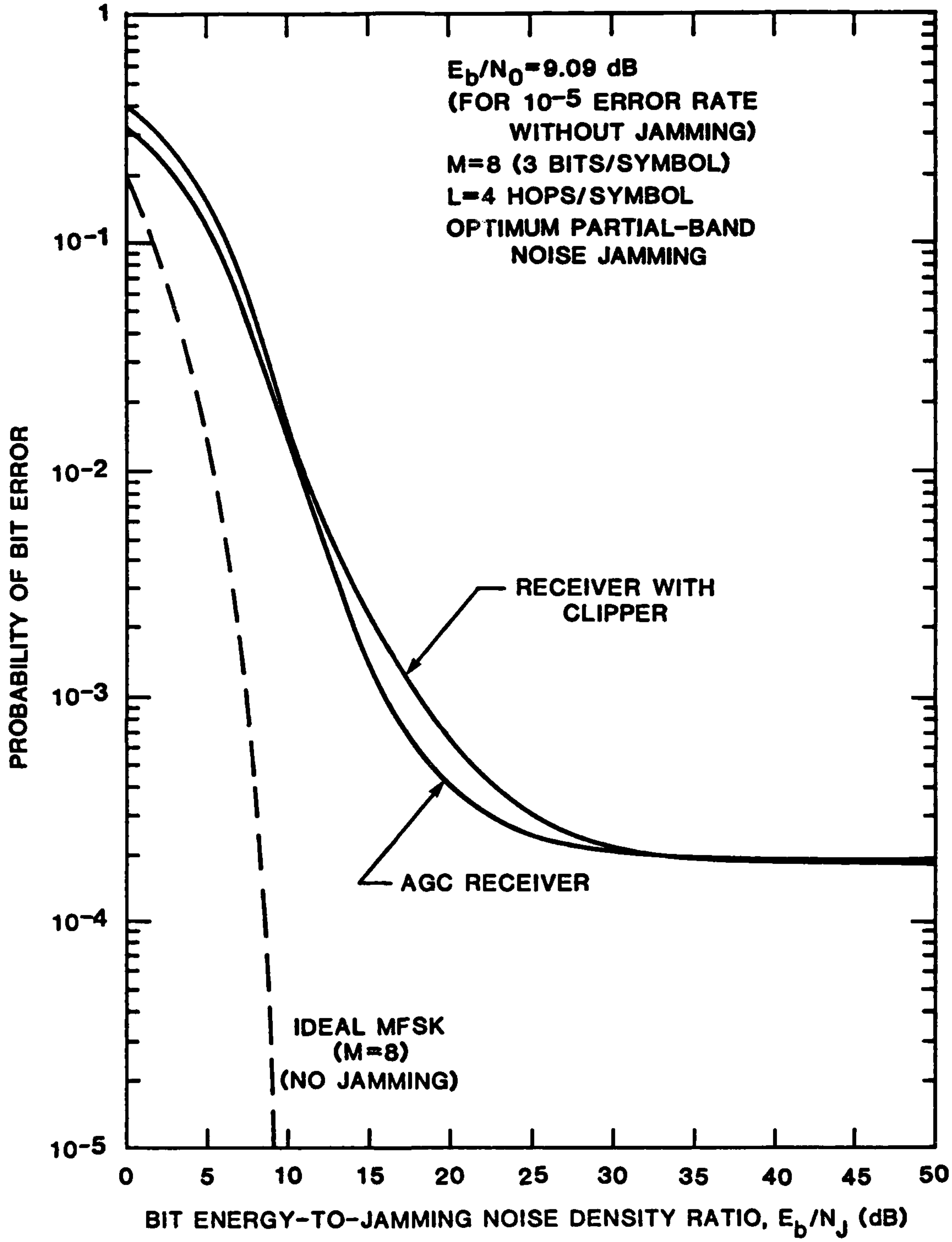
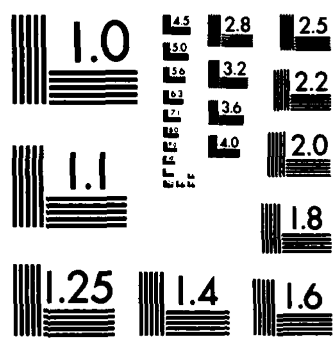


FIGURE 6-7 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/MFSK (M=8) SQUARE-LAW COMBINING RECEIVERS FOR L=4 HOPS/SYMBOL WHEN E_b/N_0=9.09 dB (FOR IDEAL MFSK (M=8) CURVE THE ABSCISSA READS E_b/N_0)

AD-A147 766 OPTIMUM JAMMING EFFECTS ON FREQUENCY-HOPPING M-ARY FSK (FREQUENCY-SHIFT K..(U) LEE (J S) ASSOCIATES INC ARLINGTON VA J S LEE ET AL. OCT 84 JC-2025-N 4/7

UNCLASSIFIED N00014-83-C-0312 F/G 17/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(single hop/symbol), it is observed that the dependence of the bit error probability upon E_b/N_J is inverse linear. As the jammer power becomes weak (E_b/N_J large), the bit error probability approaches that for thermal noise only. We observe that to produce a certain bit error rate (say 10^{-3}), E_b/N_J must be made lower (more jamming power) for the AGC and conventional receivers than for the clipper receiver. In other words, these receivers are more jammer-tolerant than the clipper receiver.

Throughout the range of E_b/N_J, the AGC receiver shows uniformly better performance than the other receivers for each value of L. But we note that the performance of the practical clipper receiver approaches that of the ideal AGC receiver for higher values of L. From the L=2 curves (Figure 6-6) we observe that the conventional receiver's performance remains inverse linear as L is increased from unity.

6.2 COMPARISONS FOR LINEAR-LAW COMBINING RECEIVERS

It was observed in the previous sections that there was very little difference in performance between the square-law and the linear-law detector schemes under worst-case partial-band noise jamming, with the linear-law case performing slightly better. A general explanation for this effect is that the envelope detector resembles a soft energy limiter which tends to suppress the jammer power more strongly than the square-law detector.

In Figures 6-8 through 6-10, the performances of the linear-law combining AGC receiver and the linear-law combining clipper receiver are compared for M=8 as a typical alphabet size. The figures are almost identical to the square-law comparison results

For both linear-law and square-law cases, the AGC receiver performs uniformly better than the clipper (soft-limiting) receiver. However, the AGC receiver model we analyzed is idealistic in the sense that perfect measurement

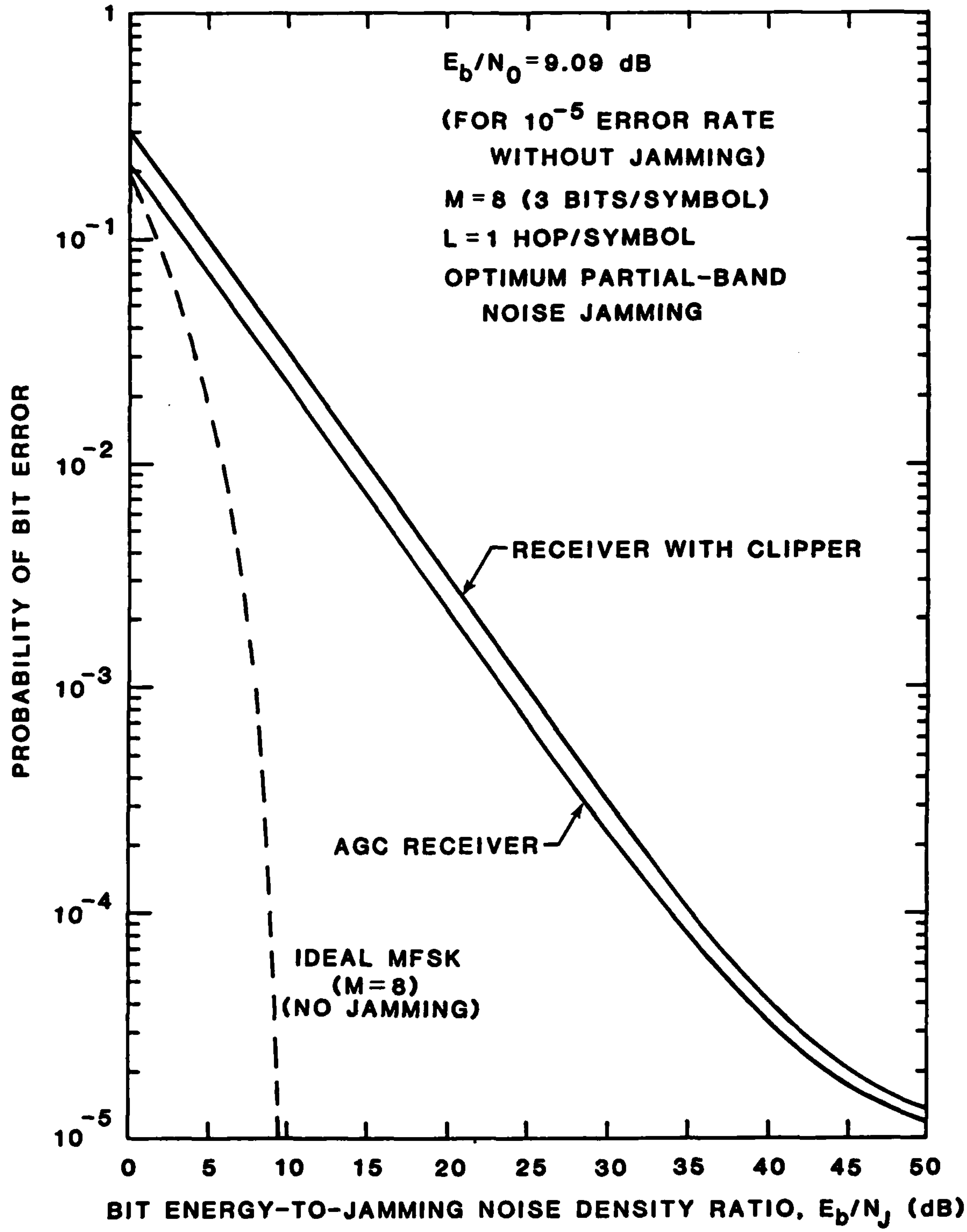


FIGURE 6-8 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/MFSK (M=8) LINEAR-LAW COMBINING RECEIVERS FOR L=1 HOP/SYMBOL WHEN E_b/N_0=9.09 dB (FOR IDEAL MFSK (M=8) CURVE THE ABSCISSA READS E_b/N_0)

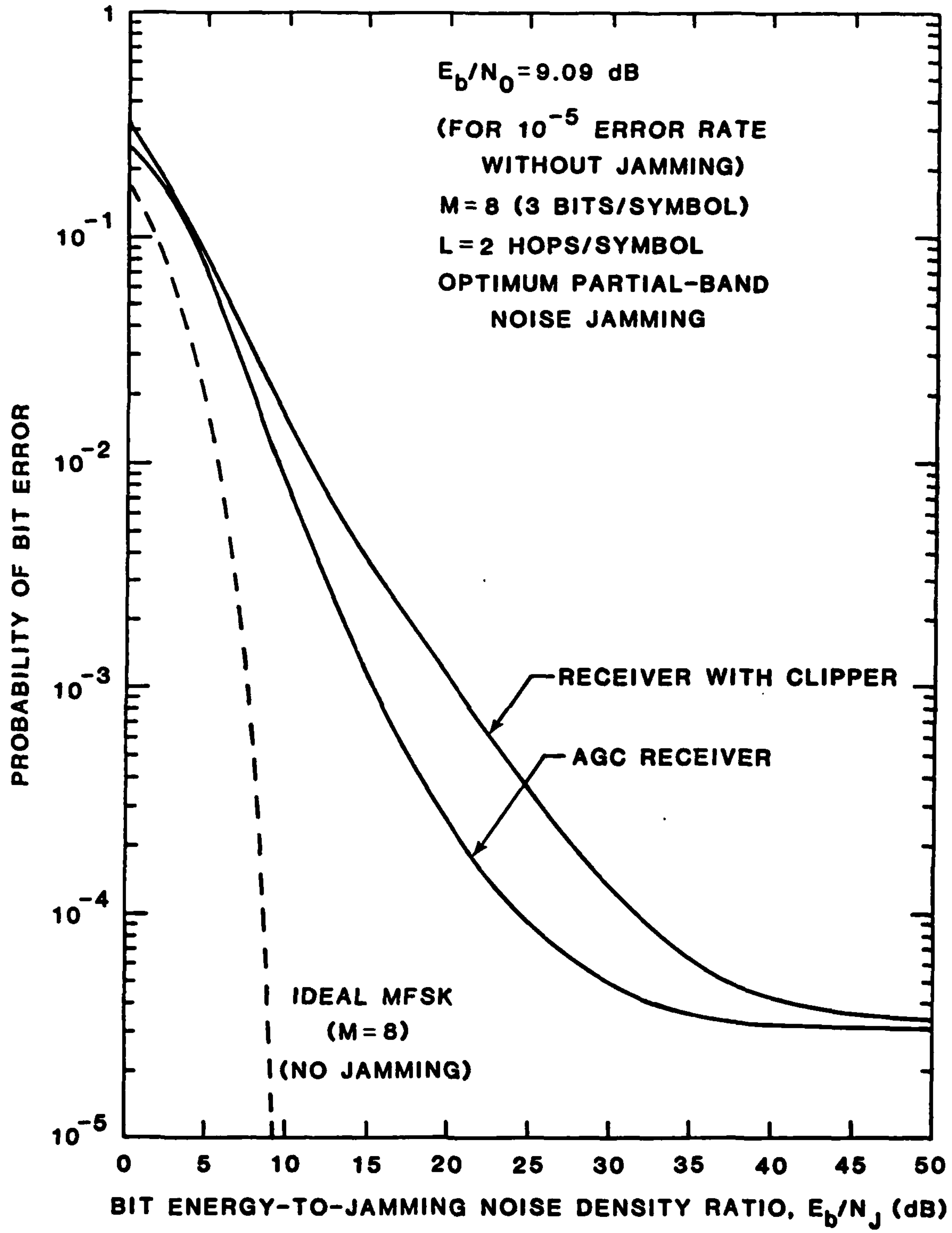


FIGURE 6-9 OPTIMUM PARTIAL-BAND NOISE JAMMING PEFORMANCE OF FH/MFSK (M=8) LINEAR-LAW COMBINING RECEIVERS FOR L=2 HOPS/SYMBOL WHEN E_b/N_0=9.09 dB (FOR IDEAL MFSK (M=8) CURVE THE ABSCISSA READS E_b/N_0)

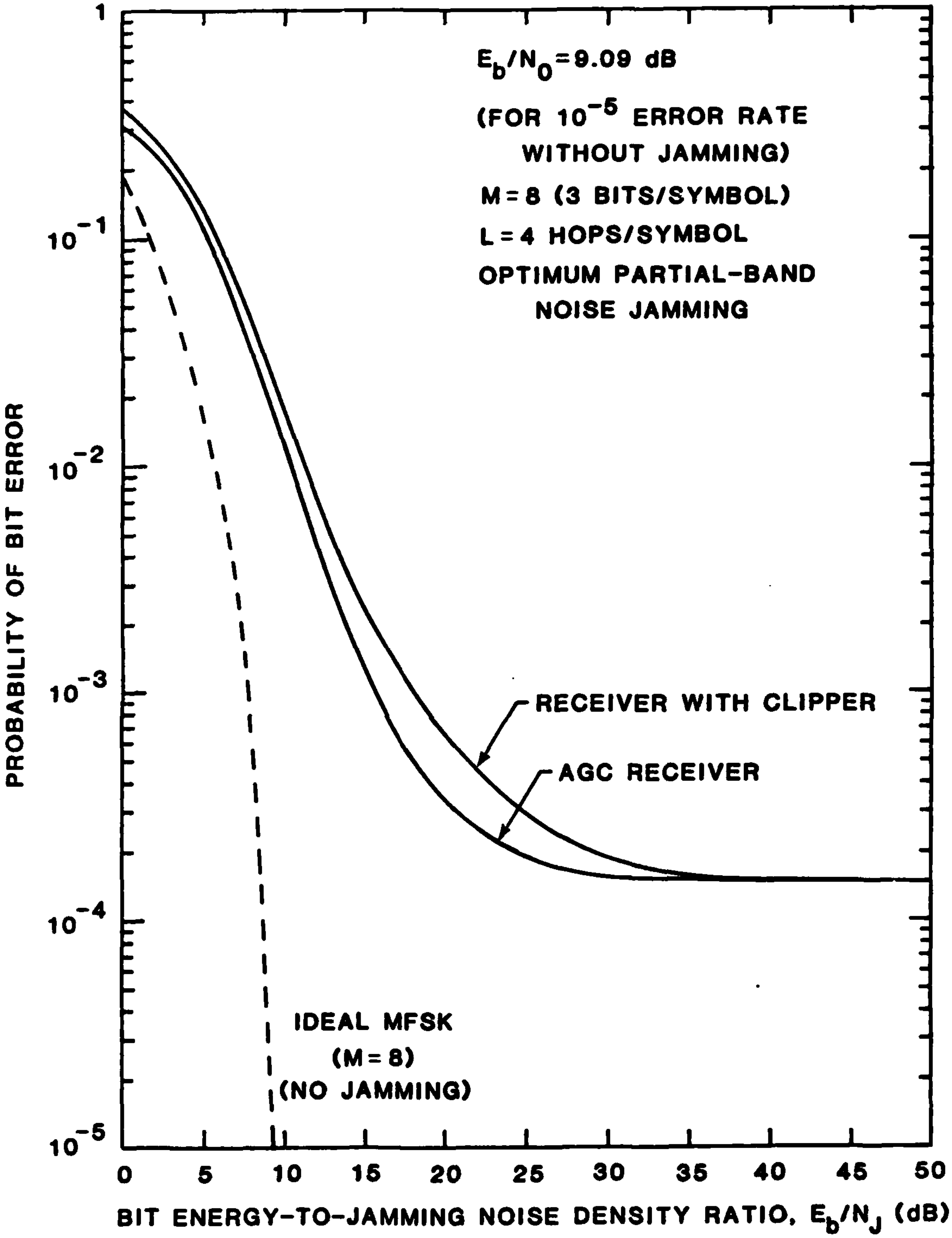


FIGURE 6-10 OPTIMUM PARTIAL-BAND NOISE JAMMING PEFORMANCE OF FH/MFSK (M=8) LINEAR-LAW COMBINING RECEIVERS FOR L=4 HOPS/SYMBOL WHEN E_b/N_0=9.09 dB (FOR IDEAL MFSK (M=8) CURVE THE ABSCISSA READS E_b/N_0)

of total noise power is assumed on each hop. Because of this ideal AGC normalization, analysis of the performance of the AGC receiver is expected to be useful as a lower bound on what might be realized in practice. On the other hand, since the clipper receiver utilizes knowledge of the thermal noise power only in setting its threshold independent of jamming information (side information), the clipper receiver could be considered a more desirable practical design.

6.3 APPLICATION OF RESULTS TO ECCM RADIO SYSTEM DESIGN

The exact results which we have obtained for the error probabilities of FH/MFSK receivers make possible a more comprehensive end-to-end analysis of a complete ECCM radio system. As suggested by Figure 6-11, such a system has several components to be designed, including error control coding, modulation, demodulation, and decoding. For the class of MFSK waveforms which we have considered, the modulation parameters include the symbol alphabet size M, the number of hops (repetitions) per symbol (L), the hop rate $R_h = 1/\tau = B$, and the total system bandwidth W. Our studies have concentrated on the performance of the system against worst-case partial-band noise jamming in terms of these parameters, and in the absence of any error control coding. In what follows we describe how the results of our analysis of the AGC square-law FH/MFSK receiver may be applied to specify the choice of L and to estimate required coding gain.

6.3.1 Selection of the Number of Hops/Symbol

It was shown in Section 4 that for given values of M, E_b/N_0, and E_b/N_J, there is an optimum value of L for which the AGC receiver's error probability is minimized. Except in the limiting case of no thermal noise, this value of L is unity for very strong jamming (small E_b/N_J), and increases as E_b/N_J increases up to a certain point, then decreases again to one as E_b/N_J

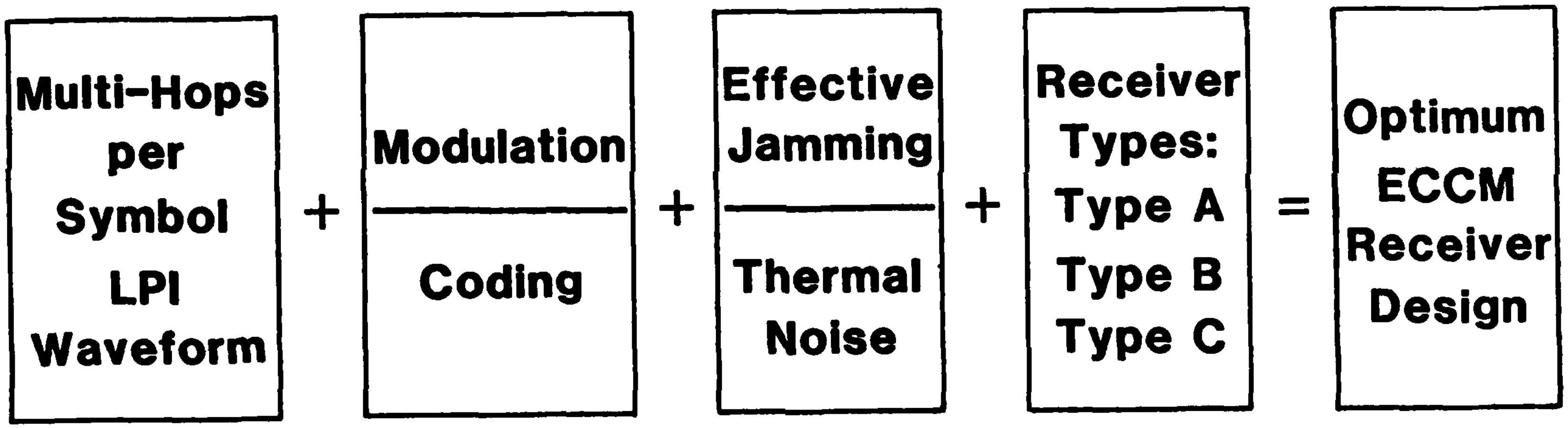


FIGURE 6-11 ECCM SYSTEM DESIGN

approaches infinity. Thus, under worst-case partial-band jamming the effect of L only partially resembles that of diversity in the fading channel, which increases indefinitely with SNR. Still, we may construct an under-envelope of a family of $P_b(e)$ vs. E_b/N_J or $P_b(e)$ vs. E_b/N_0 curves for different L to obtain a single optimum diversity curve, analogous to what is done in studies of fading.

For M=2, the under-envelopes of $P_b(e)$ vs. E_b/N_J curves for several fixed E_b/N_0 can be combined to give Figure 6-12, showing that as E_b/N_0 increases, if the optimum L is used, the error probability under optimum partial-band jamming can be made to approach within 2 dB of the ideal, unjammed performance. The values of L are shown in the figure alongside their respective segments. Very similar results hold for other values of M. In general, we may conclude from this figure that the quasi-diversity L can be used to improve the jammed performance provided that E_b/N_0 is high enough. For example, a 10^{-3} error rate can be achieved for L from 1 to 5 if E_b/N_0 is greater than 10.94 dB; the value of L increases and the value of E_b/N_J for which this error rate occurs decreases as E_b/N_0 increases. Thus, the choice of L and the resulting tolerable level of jamming are both tied to E_b/N_0.

6.3.2 Estimated Coding Gain Requirement

For a given maximum $P_b(e)$ requirement, we can also interpret the previous figure as follows: the horizontal distance between the ideal BFSK curve and a particular $P_b(e)$ vs. E_b/N_J curve represents (in dB) the amount of SNR which has to be made up or regained in order to achieve the given error rate. This concept is perhaps easier to understand if SNR is E_b/N_0 rather than E_b/N_J. In Figure 6-13, the same information as in Figure 6-12 is presented but in the form of $P_b(e)$ vs. E_b/N_0 curves for fixed E_b/N_J and optimum L.

Since plotting $P_b(e)$ vs. E_b/N_J for fixed E_b/N_0 represents what happens when the jamming power is varied, plotting $P_b(e)$ vs. E_b/N_0 for fixed E_b/N_J corres-

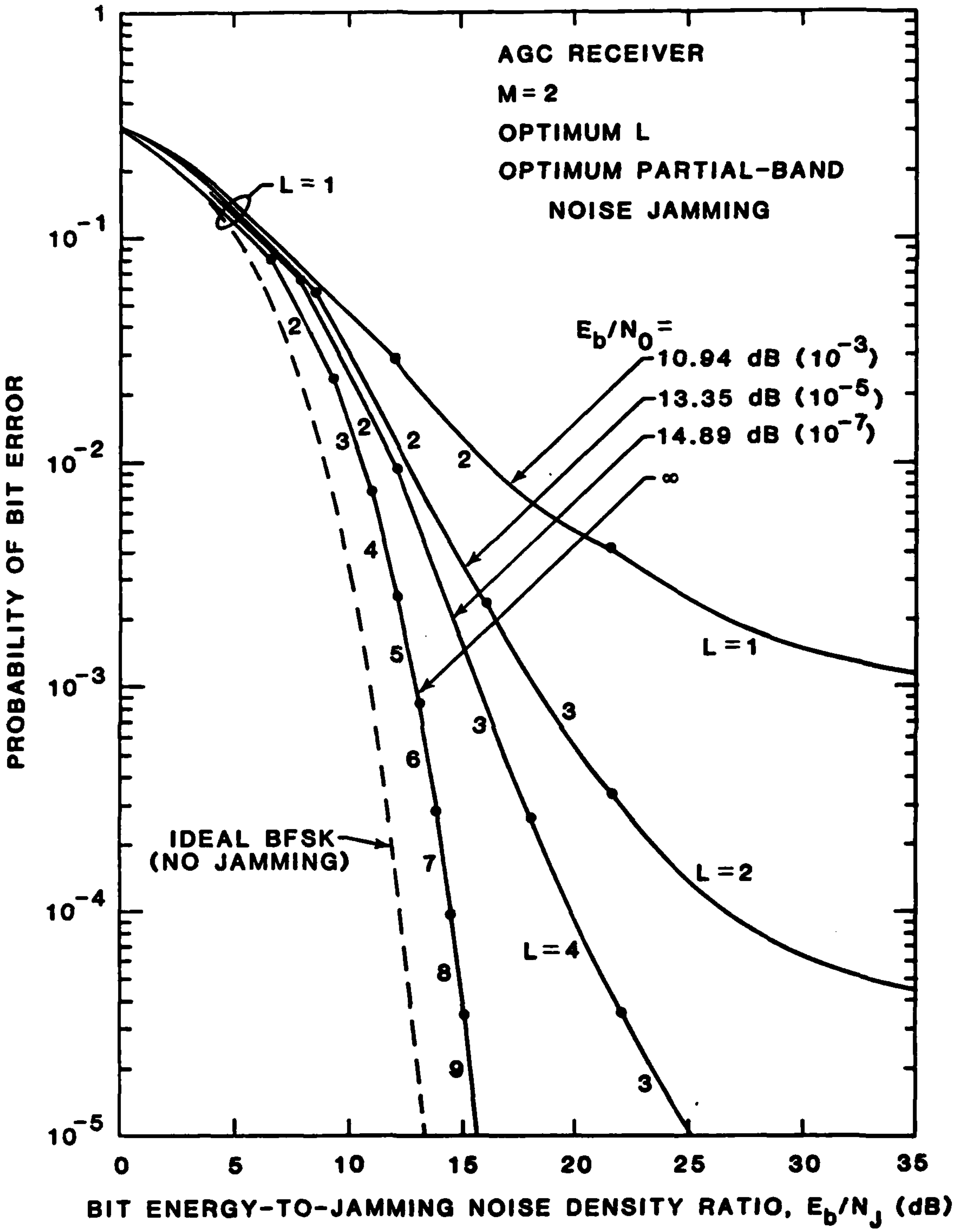


FIGURE 6-12 PROBABILITY OF BIT ERROR VS. E_b/N_J FOR AGC FH/BFSK RECEIVER USING OPTIMUM NUMBER (L) OF HOPS/BIT, FOR DIFFERENT VALUES OF E_b/N_0 IN WORST-CASE PARTIAL-BAND NOISE JAMMING

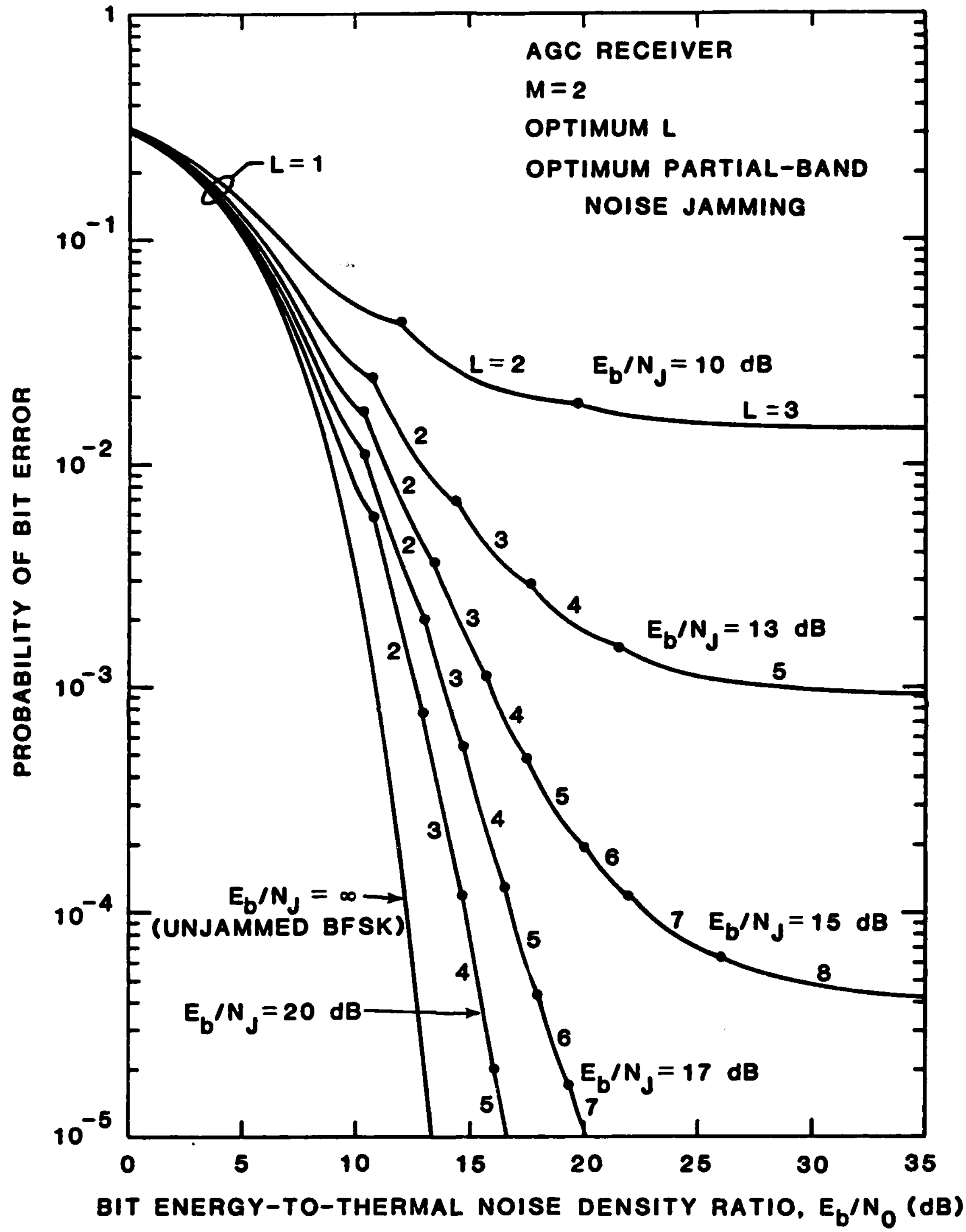


FIGURE 6-13 PROBABILITY OF BIT ERROR VS. E_b/N_0 FOR AGC FH/BFSK RECEIVER USING OPTIMUM NUMBER (L) OF HOPS/BIT, FOR DIFFERENT VALUES OF E_b/N_J IN WORST-CASE PARTIAL-BAND NOISE JAMMING

ponds physically to varying the thermal noise density rather than the transmitter power, or else to a jammer which adjusts its power proportionately as the signal power is varied. In any event, we can use Figure 6-13 to make an estimate of (a) the amount of increase in signal power necessary to maintain a fixed error rate while jammed, or (b) the amount of coding gain needed to compensate for the E_b/N_0 loss due to jamming, or (c) a combined number for increased power and coding gain, since there are limits to what coding can accomplish in this situation. The effective increases in E_b/N_0 which are necessary to maintain fixed error rates of 10^{-3} and 10^{-5} are given in Table 6-2 for M=2.

TABLE 6-2

REQUIRED SNR COMPENSATION TO MAINTAIN ERROR RATE PERFORMANCE OF FH/BFSK IN OPTIMUM PARTIAL-BAND NOISE JAMMING

DESIRED $P_b(e)$	E_b/N_J (dB)	REQUIRED COMPENSATION IN E_b/N_0 (dB)	L
10^{-3}	10	Not attainable	-
	13	17.1	5
	15	5.9	4
	17	2.9	3
	20	1.7	2
10^{-5}	10	Not attainable	-
	13	Not attainable	-
	15	Not attainable	-
	17	6.7	7
	20	3.3	5

7.0 OPTIMUM JAMMING STRATEGY AGAINST MULTI-HOPS PER SYMBOL FH/MFSK SPREAD-SPECTRUM SYSTEMS

In the previous sections, worst-case communication performances in partial-band noise jamming were analyzed in terms of the exact bit error probability expressions for different receiver types (linear combining receiver, clipper receiver, AGC receiver, and self-normalizing receiver). The worst-case performances were determined by varying the partial-band fraction γ to find the maximum bit error probability for given values of the parameters M, L, E_b/N_0, and E_b/N_J, i.e.

$$P_b(e;\gamma_0, M, L, E_b/N_0, E_b/N_J) = \max P_b(e;\gamma, M, L, E_b/N_0, E_b/N_J). \tag{7-1}$$

In the process of finding the worst-case communications performance, we have in fact determined the specifications of an optimal partial-band noise jammer in terms of γ and the ratio E_b/N_J at the receiver. It is assumed that the jammer's total power J as observed at the receiver is correctly placed so as to lie entirely within the W Hz hopping system bandwidth. In general, the jammer's optimum partial-band jamming fraction γ_0 is a different function of M, L, E_b/N_0, and E_b/N_J for each receiver type:

$$\gamma_0 = \gamma_{0R}(M, L, E_b/N_0, E_b/N_J) \tag{7-2}$$

where the subscript R denotes receiver type.

In this section we consider how to apply what we have learned about optimum jamming of FH/MFSK systems from the performance analyses to practical aspects of jammer system design. First, we discuss some of the basic issues affecting the selection of jamming parameters; then we assess the sensitivity of the jammer's effectiveness to departures of these parameters from optimum

values. Finally, we show how this information may be used to configure a conceptual jammer system design.

7.1 BASIC JAMMING SYSTEM CONSIDERATIONS

For the continuously emitted noise type of jamming we are studying, there are two parameters under the jammer's direct control: γ, the fraction of the bandwidth W which is to be jammed; and J, the transmitted jamming power. These two parameters are to be chosen so that at the receiver certain values of γ and E_b/N_J are achieved. Or, it may be the jammer has sufficient power to opt for wideband jamming (γ=1) and still be assured that the targeted communications system is significantly degraded. The factors which are involved may be classified as scenario-dependent and as receiver-dependent.

7.1.1 Scenario-Dependent Factors

In an electronic warfare environment the communicator and the jammer have conflicting objectives. The objective of the communicator is to utilize a communications waveform which achieves a low probability of intercept (LPI) and to employ receiver processing which, if jammed as a result of the signal's being detected, would mitigate the effects of the interference. The FH/MFSK spread-spectrum (SS) waveforms are power efficient and therefore fulfill LPI waveform design goals. To improve the degree of covertness, the designer may employ a multi-hops/symbol strategy to further weaken the energy density of the transmitted signal per hop. Two possible scenarios are depicted in Figures 7-1 and 7-2. Figure 7-1 shows a communication link from a ground transmitter to an airborne receiver, employing an FH/MFSK SS system, while Figure 7-2 depicts a satellite communication system using the same modulation format.

The classical way of viewing the interaction between communications and jamming for spread spectrum systems is in terms of a power battle. The

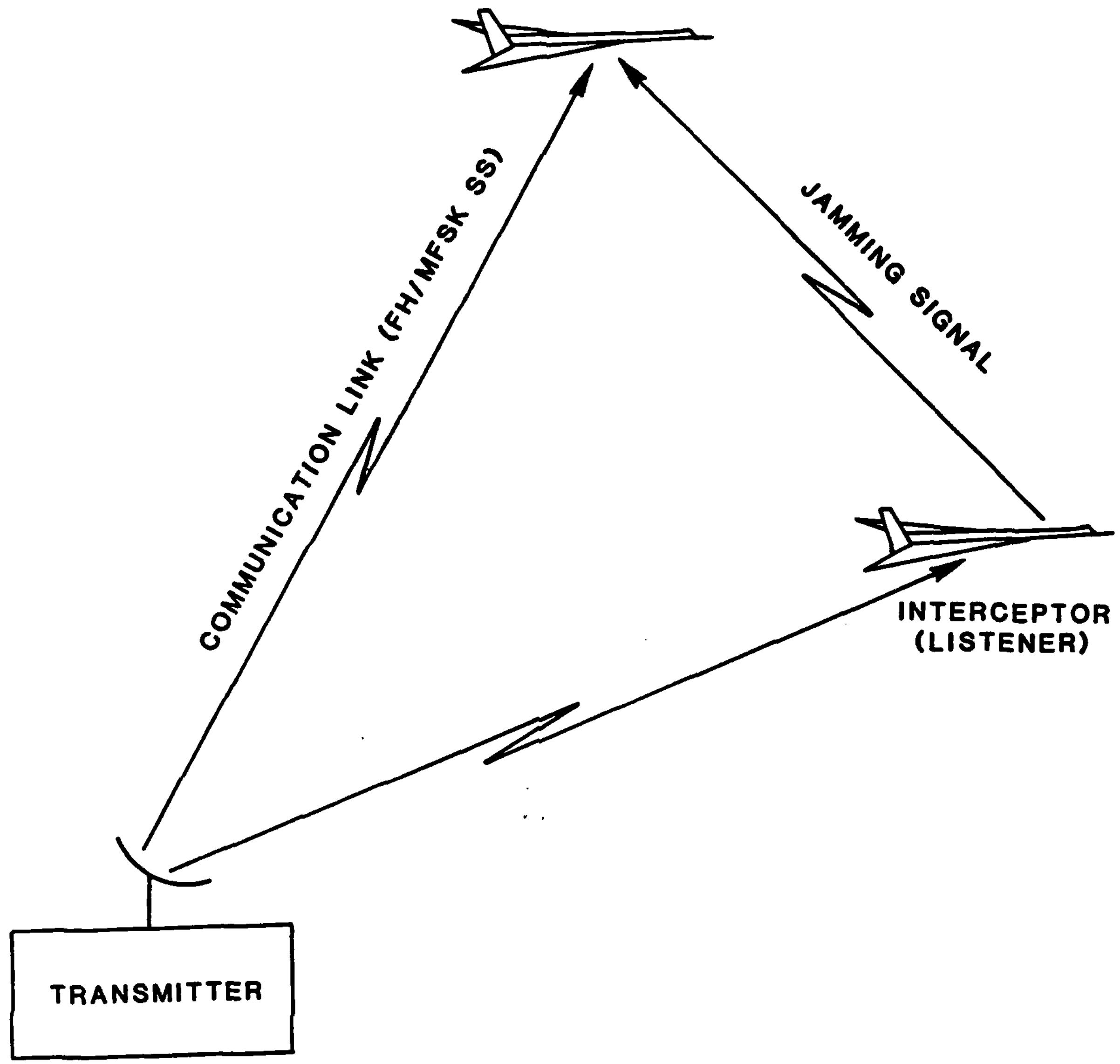


FIGURE 7-1 FREQUENCY HOPPING MFSK SPREAD-SPECTRUM LPI COMMUNICATION IN EW ENVIRONMENT (GROUND-TO-AIR COMMUNICATION LINK)

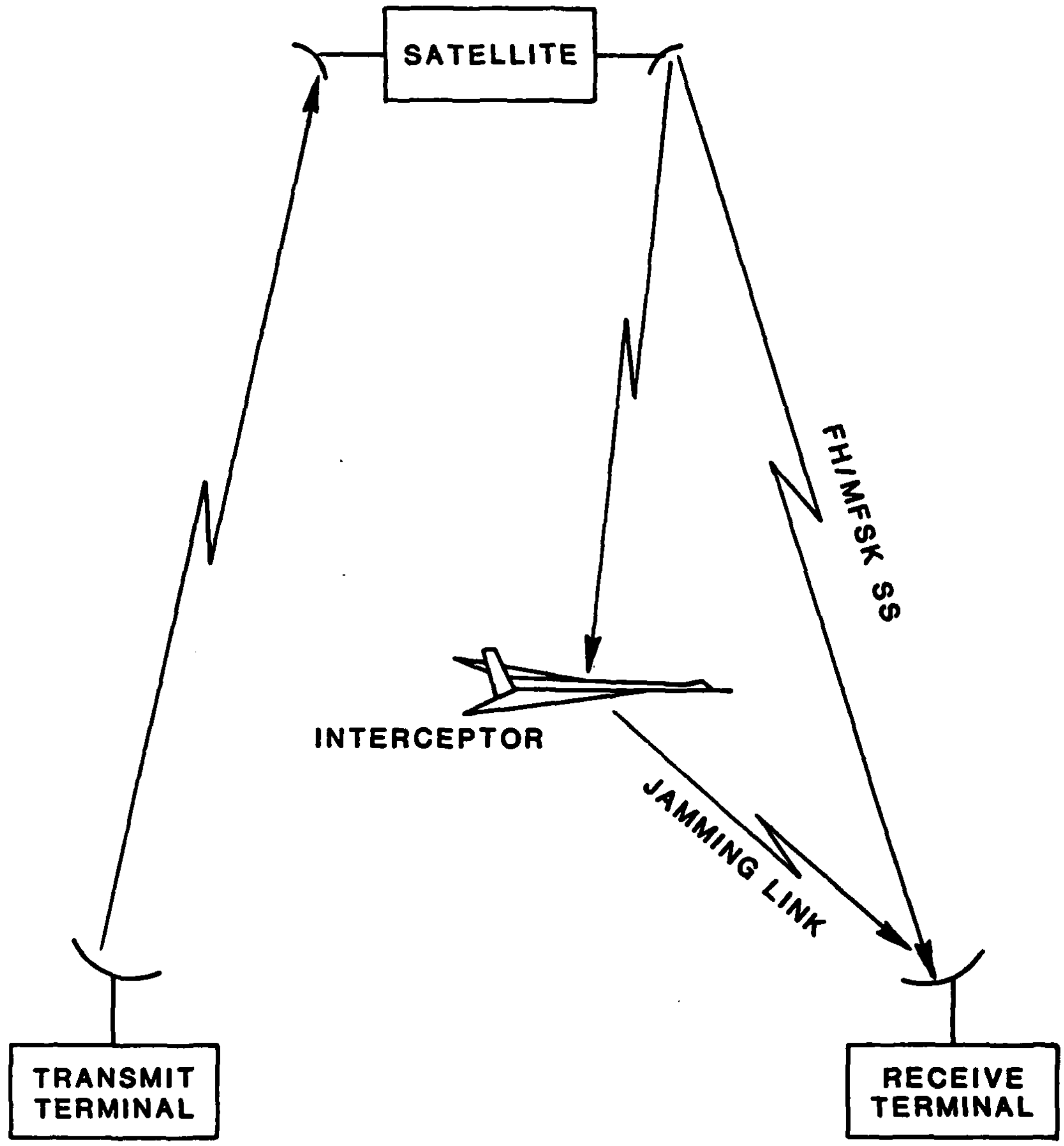


FIGURE 7-2 FREQUENCY HOPPING MFSK SPREAD-SPECTRUM LPI COMMUNICATION IN EW ENVIRONMENT (GROUND-SATELLITE-GROUND COMMUNICATION LINK)

received bit energy is

$$E_b = \frac{E_s}{K} = \frac{L}{K} E_h = \frac{L}{K} \cdot \frac{S}{B} \propto \frac{L}{K} \cdot \frac{S_0}{B} \cdot \frac{G_T}{R_T^2} \tag{7-3a}$$

where

$$\left.\begin{aligned}
E_s &= \text{received symbol energy} \\
E_h &= \text{received hop energy} \\
L &= \text{number of hops per symbol} \\
K &= \log_2 M = \text{number of bits per symbol} \\
S &= \text{received signal power} \\
S_0 &= \text{transmitted signal power*} \\
G_T &= \text{receiver antenna gain in direction of transmitter} \\
R_T &= \text{receiver-to-transmitter range} \\
B &= R_h = \text{hop rate.}
\end{aligned}\right\} \tag{7-3b}$$

The received jamming noise power spectral density is given by

$$N_J = \frac{J}{W} \propto \frac{J_0}{W} \cdot \frac{G_J}{R_J^2} \tag{7-4a}$$

where

$$\left.\begin{aligned}
J &= \text{received jammer noise power} \\
W &= \text{spread spectrum bandwidth} \\
J_0 &= \text{transmitted jammer noise power*} \\
G_J &= \text{receiver antenna gain in direction of jammer} \\
R_J &= \text{receiver-to-jammer range.}
\end{aligned}\right\} \tag{7-4b}$$

*EIRP in the direction of the receiver.

Thus the received bit energy-to-jammer noise density ratio E_b/N_J is given by

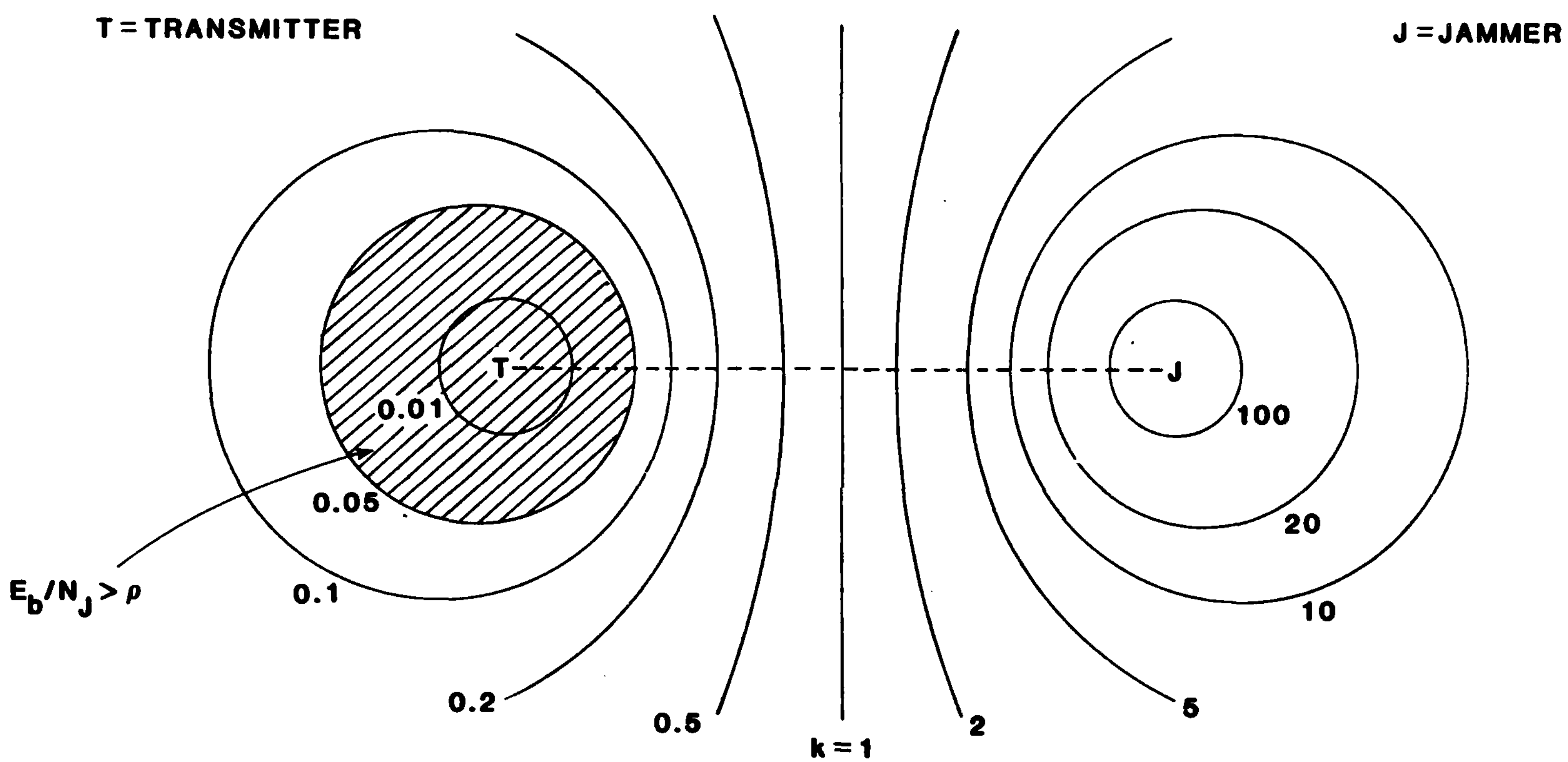
$$\frac{E_b}{N_J} = \frac{L}{K} \cdot \frac{W}{B} \cdot \frac{S}{J} = \frac{L}{K} \cdot \frac{W}{B} \cdot \frac{S_0}{J_0} \cdot \left(\frac{R_J}{R_T}\right)^2 \frac{G_T}{G_J}; \tag{7-5}$$

and if we consider the requirement $E_b/N_J > \rho$, we obtain the equation

$$\left(\frac{R_T}{R_J}\right)^2 < \frac{1}{\rho} \cdot \frac{L}{K} \cdot \frac{W}{B} \cdot \frac{S_0}{J_0} \cdot \frac{G_T}{G_J} = k. \tag{7-6}$$

If the dependence of G_T and G_J on position is ignored, then (7-6) describes a sphere inside of which $E_B/N_J > \rho$ for $k < 1$ and outside of which $E_b/N_J > \rho$ for $k > 1$. Figure 7-3 shows a section of the family of spheres for different values of k. This type of display has been used to illustrate the advantages of spread-spectrum systems in combatting wideband noise jamming. For example, for conventional BFSK, L=K=1 and W=B, if we ignore antenna considerations, then the requirement $E_b/N_J > 10$ dB gives $k = S_0/10J_0$, or, in terms of allowable J_0/S_0 (jamming margin), $k = 0.1/(J_0/S_0)$. A spread-spectrum bandwidth $W = 10^3B$ gives $k = 100/(J_0/S_0)$. In Figure 7-4 we see that effective conventional (narrowband) communication is restricted to receivers near the communications transmitter, while spread spectrum communication is effective in this example everywhere except quite near the jammer.

In this type of analysis which leads to the traditional view of spread spectrum system performance as illustrated by Figure 7-4, it is assumed that the entire system bandwidth W is jammed and that received jamming power is sufficiently large that thermal noise may be neglected. For these assumptions it is reasonable to consider the bandwidth ratio W/B as a processing gain. For partial-band jamming of frequency hopping systems, it might seem reasonable to consider the processing gain to be the ratio $\gamma W/B$, in which case



$$k = \frac{1}{\rho} \cdot \frac{L}{K} \cdot \frac{W}{B} \cdot \frac{S_0}{J_0} \cdot \frac{G_T}{G_J}$$

FIGURE 7-3 CONTOURS GENERATED BY EQUATION (7-6)

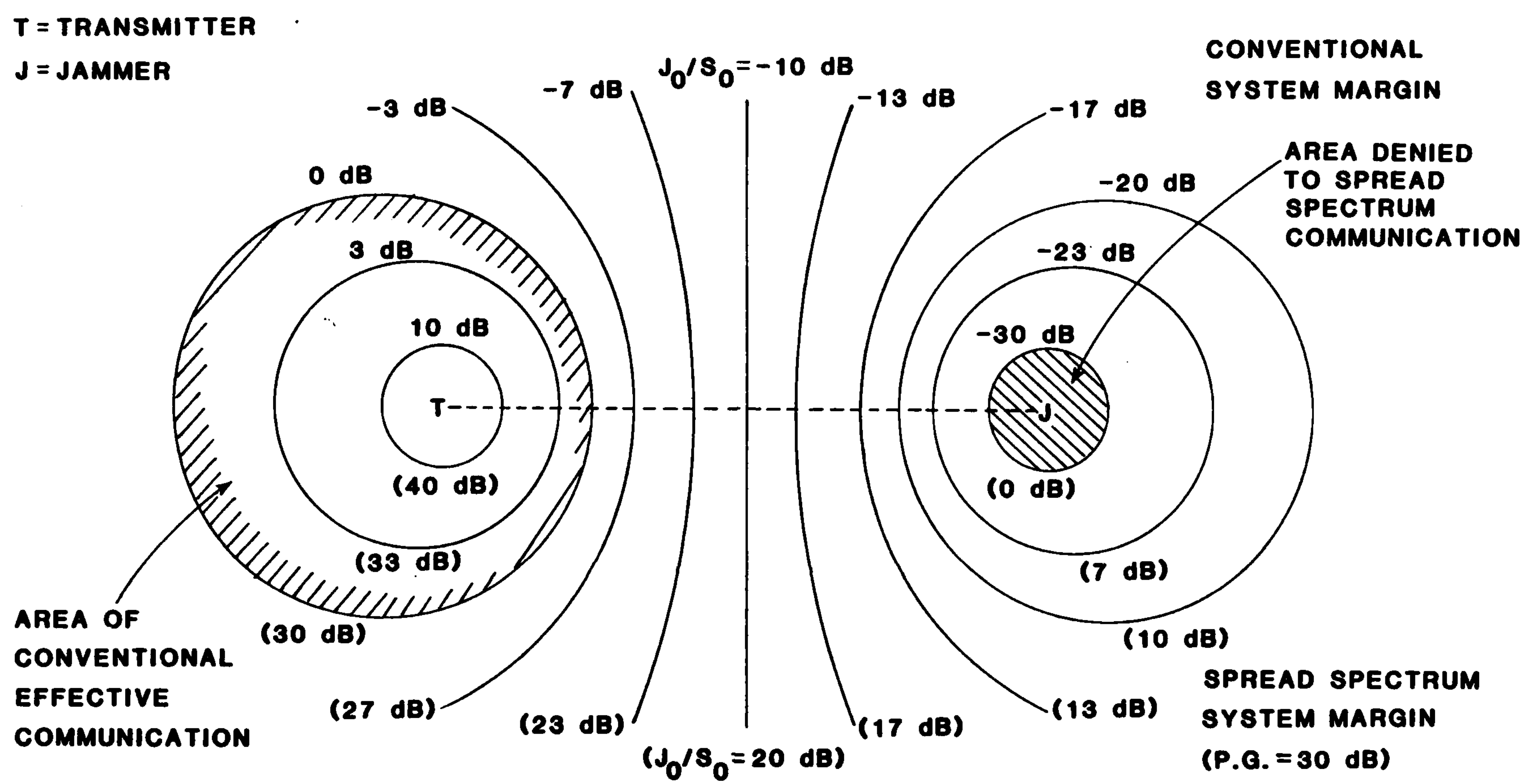


FIGURE 7-4 COMPARISON OF AREA COVERAGES FOR CONVENTIONAL AND SPREAD SPECTRUM COMMUNICATION SYSTEMS

the $E_b/N_J = 10$ dB contours are as shown in Figure 7-5, parametric on γ for $W/B = 10^3$ and $S_0/J_0 = 1$ ($k = 100\gamma$). As illustrated by the figure, the jammer could overcome the wideband processing gain simply by making γ smaller. However, this interpretation is faulty because it does not take into account the hops which are not jammed nor any receiver processing utilizing multiple hops per symbol to make the decision. The notion of processing gain does not apply in a straightforward way to partial-band jamming. Curves such as Figure 7-5 are useful, however, for depicting the geometry the jammer must take into account to achieve a given combination of γ and E_b/N_J at the receiver.

The jammer must estimate received signal and jamming noise powers; this requires some knowledge of the relative locations of transmitter, receiver, and jammer as well as transmitted signal power. In scenarios similar to those shown in Figures 7-1 and 7-2, it may be reasonable for the jammer to consider the signal power measured at his location to be within a few dB of that at the intended receiver; in this case, the jammer-to-receiver distance, propagation factors, and antenna characteristics are the information he requires to correctly adjust his radiated power.

7.1.2 Receiver-Dependent Factors

It has been demonstrated in the analysis sections that the optimum value of γ is dependent upon the MFSK alphabet size M, the received signal bit energy-to-thermal noise density ratio E_b/N_0, and the received signal bit-energy-to-jamming noise density ratio E_b/N_J. Assuming that these parameters are known or estimated, there still remains the dependency of γ on L, the number of hops/symbol, and the specific receiver type.

Depending on receiver type and L, the jammer may elect to perform wideband jamming if sufficient power is available. The issue is well illustrated by Figures 7-6 and 7-7 which are plots of the regions of (E_b/N_0, E_b/N_J) for

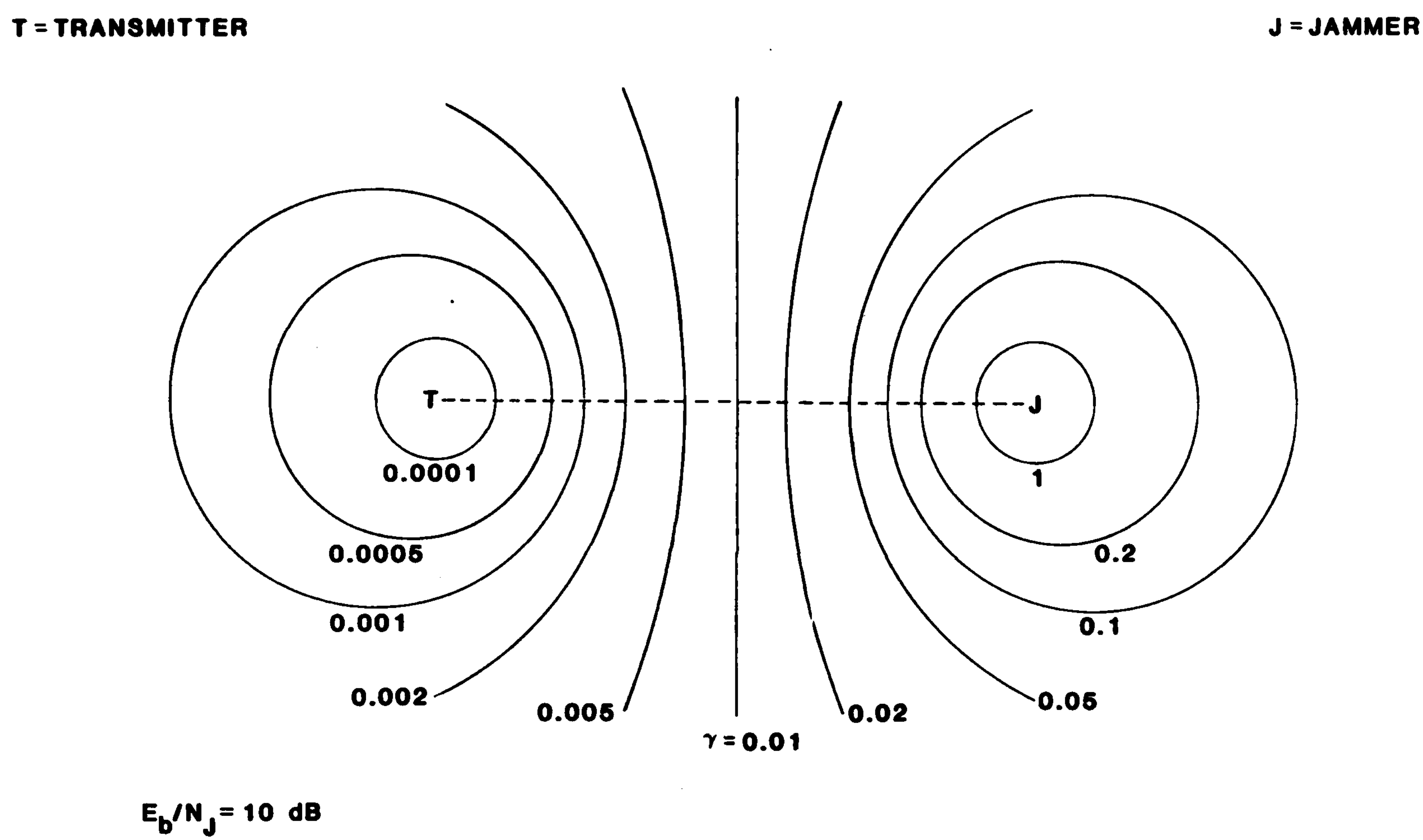


FIGURE 7-5 INFLUENCE OF PARTIAL-BAND JAMMING FRACTION (γ) ON FH COMMUNICATIONS COVERAGE (WHEN JAMMED)

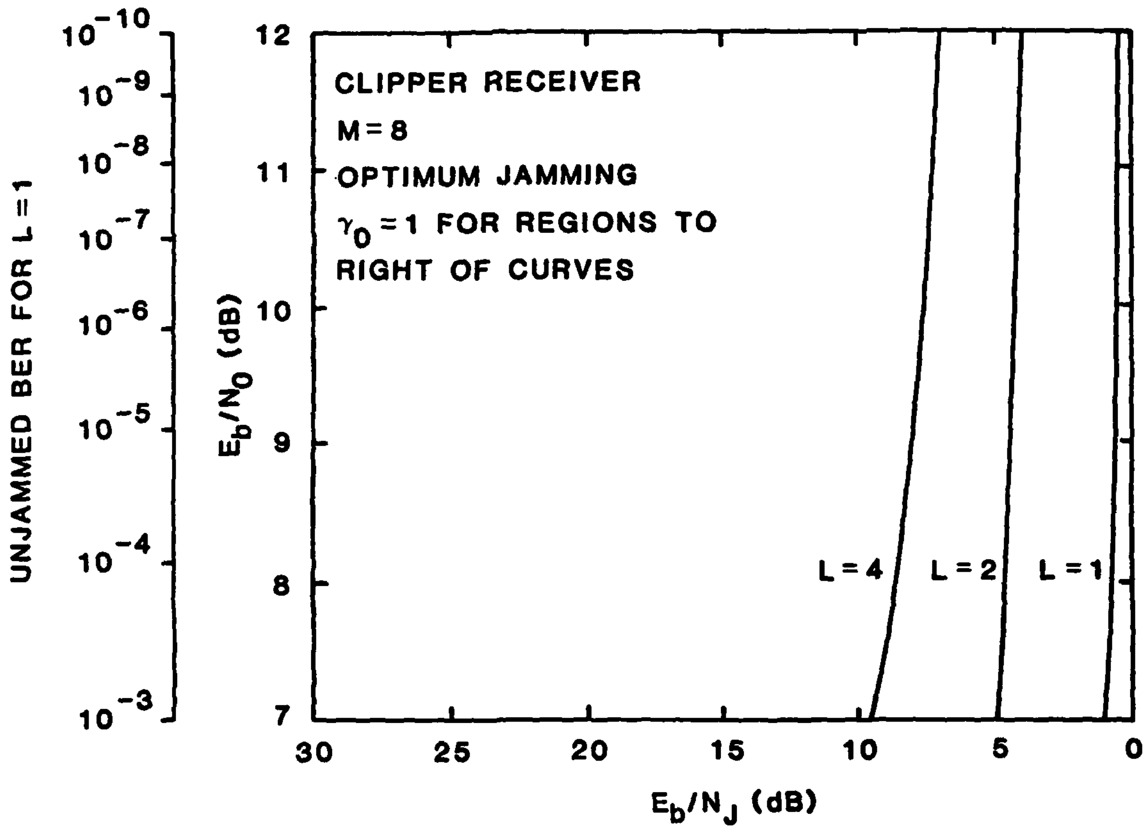


FIGURE 7-6 CONDITIONS FOR WHICH $\gamma_0 = 1$ FOR THE CLIPPER RECEIVER (M = 8)

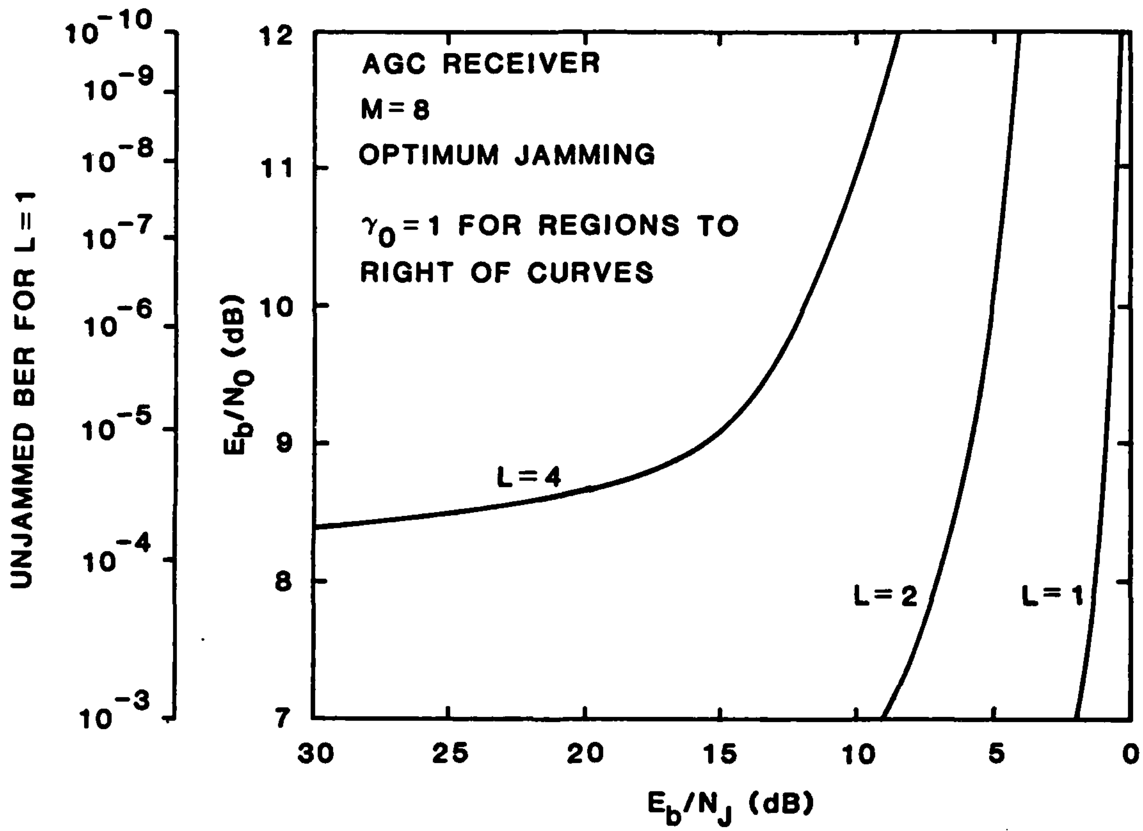


FIGURE 7-7 CONDITIONS FOR WHICH $\gamma_0 = 1$ FOR THE AGC RECEIVER (M=8)

which the optimum γ becomes unity, for different values of L. Figure 7-6 shows for the clipper receiver and M=8 that the choice between partial-band and wideband jamming is relatively insensitive to the value of E_b/N_0; as long as the jammer has enough power to insure that E_b/N_J is less than some value (a function of L), then wideband jamming is optimum.

In constrast, Figure 7-7 shows for the AGC receiver and M=8 that the choice between partial-band and wideband jamming depends significantly on the value of E_b/N_0 for $L \geq 2$. We observe that γ_0 for the AGC receiver becomes equal to unity at a smaller amount of jamming power than for the clipper receiver. The optimum value of γ depends strongly on receiver type as a function of L. For example, for M=8 and $E_b/N_0 = 9.09$ dB, in Figure 7-8 γ_0 is plotted vs. L for different values of E_b/N_J for both the clipper and the AGC receivers. It is evident that γ_0 is more sensitive to L for the AGC receiver.

7.2 SENSITIVITY OF JAMMING EFFECTS TO ERRORS IN SELECTION OF PARAMETERS

With the knowledge of the communicator's receiver type, the alphabet size M, and the number of hops/symbol L through intelligence, the jammer's optimum strategy is based on the probability of error expression as a function of three parameters: the optimum fraction γ_0, the bit energy-to-noise density ratio E_b/N_0, and the bit energy-to-jamming noise density ratio E_b/N_J:

$$P_b(e) = P_b(e;\ \gamma_0,\ E_b/N_0,\ E_b/N_J). \tag{7-7}$$

Among the three parameters, let us assume the jammer knows the E_b/N_0 value. In some cases, the thermal noise density N_0 ($N_0 = kT$, where k is Boltzmann's constant) is available to a jammer through intelligence. Then the error probability expression only depends on the optimum fraction γ_0 and E_b/N_J (mainly on the jamming power, J).

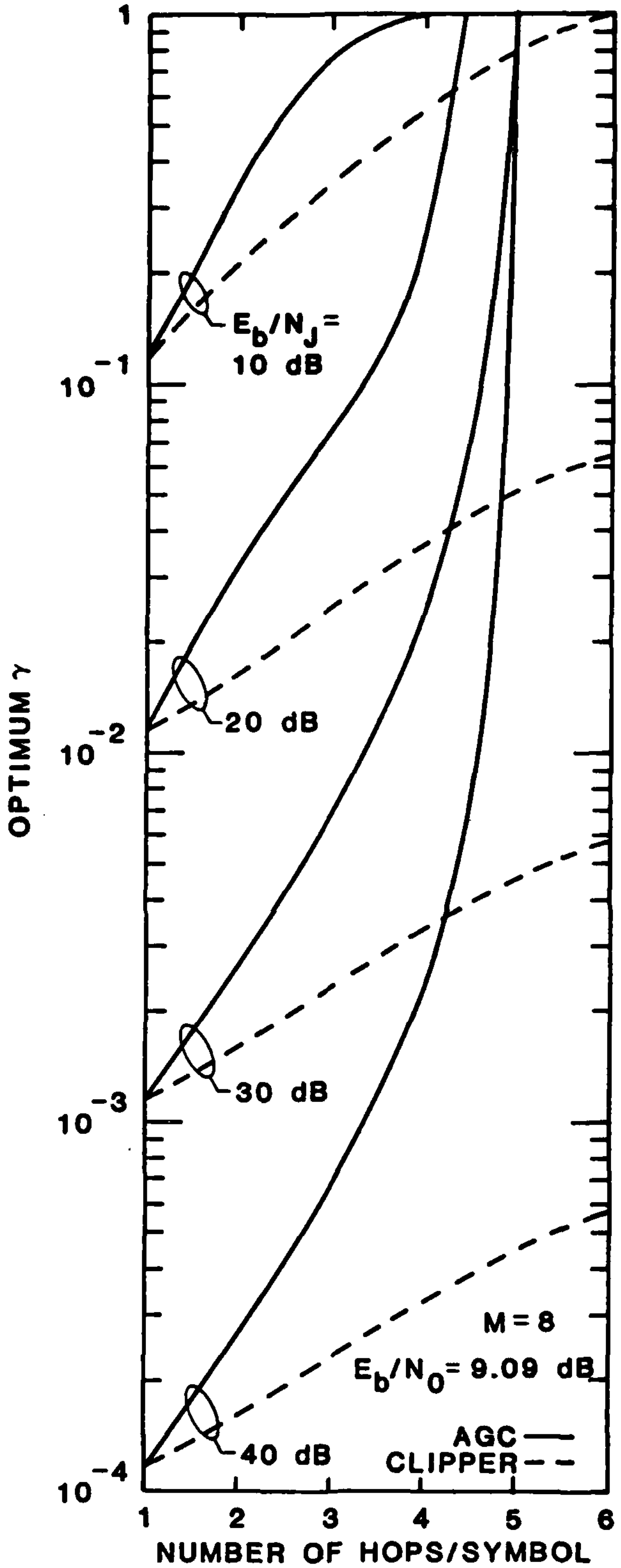


FIGURE 7-8 OPTIMUM JAMMING FRACTION (γ) VS. NUMBER OF HOPS/SYMBOL (L) FOR THE AGC AND CLIPPER FH/MFSK (M = 8) RECEIVERS WHEN $E_b/N_0 = 9.09$ dB WITH E_b/N_J AS A PARAMETER (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

Against a known receiver type, to design the best jammer, it is worthwhile to investigate how departures from the optimum values of each of the parameters γ_0 and E_b/N_J affect the optimum partial-band noise jamming performance. It is shown below that an acceptable tolerance in the jammed BER can be maintained for achievable tolerances in γ and E_b/N_J.

7.2.1 Sensitivity of the Jammed Error Rate to γ_0

The optimum partial-band jamming fraction γ_0 is the value of γ for which the probability of error is maximized for given values of E_b/N_0, E_b/N_J, M, L, and receiver type. A typical example of the manner in which the error rate depends on γ is shown in Figure 7-9 for the AGC receiver with M=4, L=1, and $E_b/N_0 = 10.61$ dB. For each value of E_b/N_J shown in the figure, the error rate is a unimodal function of γ. Because γ is constrained to be in the interval [0,1], for $E_b/N_J = 0$ dB the optimum γ is taken to be $\gamma_0 = 1$.

We may determine the sensitivity of the jammer's effect on the receiver to an incorrect choice of γ by the following method. Let the nominal error rate be

$$P_0 = P_b(e;\ \gamma_0,\ E_b/N_J,\ E_b/N_0,\ L,\ M); \tag{7-8}$$

for example, in Figure 7-9 for $E_b/N_J = 20$ dB, $P_0 = 2.7 \times 10^{-3}$ for $\gamma_0 = 1.4 \times 10^{-2}$. As shown in the figure, if we accept an achieved error rate of $P_0 - \Delta P \leq P_b(e) \leq P_0$, then for fixed E_b/N_J we can tolerate variations in γ such that $\gamma_{0min} \leq \gamma \leq \gamma_{0max}$. Thus the sensitivity to γ_0 can be expressed as the percentage of variation in γ for which a given relative performance degradation $\Delta P/P_0$ is maintained.

Tables 7-1 and 7-2 give typical sensitivities of the jammed clipper receiver's performance to γ_0 for M=2 and M=8, respectively. The same information for the AGC receiver is presented in Tables 7-3 and 7-4. Inspection of these data reveals that for either receiver type, the jammer can accomplish nearly

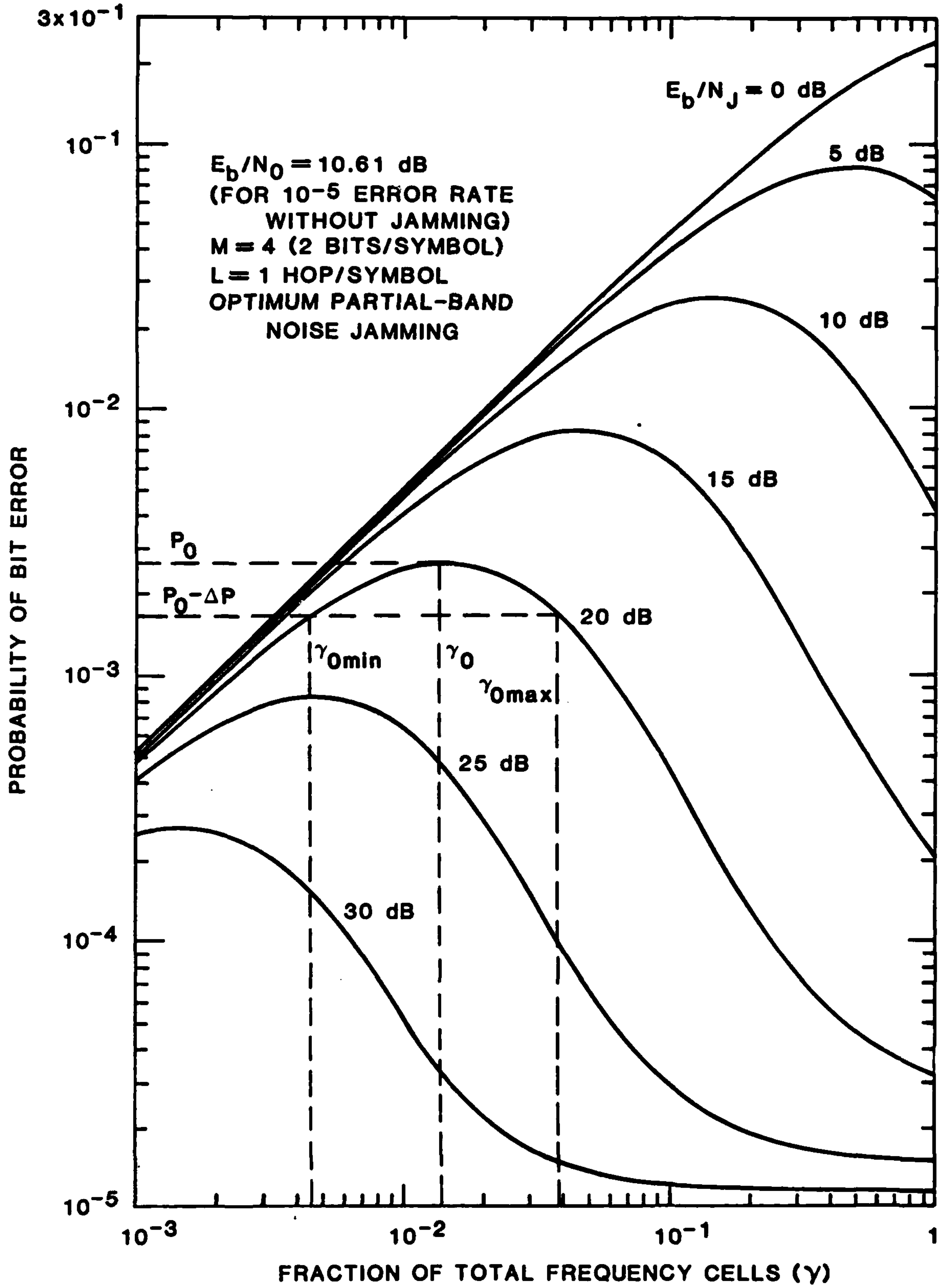


FIGURE 7-9 SENSITIVITY OF PROBABILITY OF BIT ERROR TO INCORRECT CHOICE OF γ FOR FH/MFSK (M=4) AGC LINEAR-LAW RECEIVER WITH L=1 HOP/SYMBOL WHEN E_b/N_0=10.61 dB

TABLE 7-1

SENSITIVITY OF JAMMED BER TO γ_0 FOR THE CLIPPER SQUARE-LAW RECEIVER (M=2)

L	E_b/N_0 (dB)*	E_b/N_J (dB)	γ_0 / P_0	$\Delta P/P_0$ = 10%		$\Delta P/P_0$ = 5%	
				γ_{0min}	γ_{0max}	γ_{0min}	γ_{0max}
1	10.94	15	9.0(-2) / 1.67(-2)	4.93(-2) (-45.2%)	1.57(-1) (74.4%)	5.93(-2) (-34.1%)	1.275(-1) (41.7%)
		30	3.0(-3) / 1.46(-3)	$<10^{-3}$	8.32(-3) (177%)	1.37(-3) (-54.3%)	5.77(-3) (92.3%)
	13.35	15	8.0(-2) / 1.57(-2)	4.65(-2) (-41.9%)	1.21(-1) (51.3%)	5.47(-2) (-31.6%)	1.06(-1) (32.5%)
		30	2.0(-3) / 4.98(-4)	1.57(-3) (-21.5%)	4.08(-3) (104%)	1.82(-3) (-9%)	3.53(-3) (76.5%)
4	10.94	15	7.0(-1) / 2.59(-2)	2.09(-1) (-70.1%)	1.0 (42.9%)	2.83(-1) (-59.6%)	1.0 (42.9%)
		30	2.0(-2) / 6.89(-3)	$<10^{-3}$	1.0 (4900%)	1.03(-2) (-48.5%)	1.0 (4900%)
	13.35	15	4.0(-1) / 7.0(-1)	2.52(-1) (-37%)	9.2(-1) (130%)	2.89(-1) (-27.8%)	7.08(-1) (77%)
		30	9.0(-3) / 2.3(-4)	3.05(-3) (-66.1%)	4.37(-2) (386%)	4.3(-3) (-52.2%)	2.18(-2) (142%)

*E_b/N_0 for L=1 BER's of 10^{-3} (10.94 dB) and 10^{-5} (13.35 dB)

Note: $a(-n) \equiv a \times 10^{-n}$

TABLE 7-2

SENSITIVITY OF JAMMED BER TO γ_0 FOR THE CLIPPER SQUARE-LAW RECEIVER (M=8)

L	E_b/N_0 (dB)*	E_b/N_J (dB)	γ_0 P_0	$\Delta P/P_0$ = 10%		$\Delta P/P_0$ = 5%	
				γ_{0min}	γ_{0max}	γ_{0min}	γ_{0max}
1	6.97	15	4.11(-2) 1.03(-2)	2.4(-2) (-41.6%)	7.25(-2) (76.4%)	2.76(-2) (-32.8%)	6.2(-2) (50.9%)
		30	1.3(-3) 1.32(-3)	$<10^{-3}$	5.3(-3) (308%)	$<10^{-3}$	3.19(-3) (145%)
	9.09	15	3.74(-2) 9.54(-3)	2.38(-2) (-36.4%)	5.76(-2) (54%)	2.69(-2) (-28.1%)	5.18(-2) (38.5%)
		30	1.18(-3) 3.12(-4)	$<10^{-3}$	1.8(-3) (52.5%)	$<10^{-3}$	1.46(-3) (23.7%)
4	6.97	15	2.0(-1) 1.7(-2)	6.74(-2) (-66.3%)	1.0 (400%)	9.01(-2) (-55.0%)	1.0 (400%)
		30	5.51(-3) 7.98(-3)	$<10^{-3}$	1.0 (18048%)	$<10^{-3}$	1.0 (18048%)
	9.09	15	1.33(-1) 2.36(-3)	7.75(-2) (-41.7%)	2.56(-1) (92.5%)	8.99(-2) (-32.4%)	2.07(-1) (55.6%)
		30	3.31(-3) 2.22(-4)	$<10^{-3}$	1.0 (30111%)	1.22(-3) (-63.1%)	1.61(-2) (386%)

*E_b/N_0 for L=1 BER's of 10^{-3} (6.97 dB) and 10^{-5} (9.09 dB)

Note: $a(-n) \equiv a \times 10^{-n}$

TABLE 7-3

SENSITIVITY OF JAMMED BER TO γ_0 FOR THE AGC SQUARE-LAW RECEIVER (M=2)

L	E_b/N_0 (dB)*	E_b/N_J (dB)	γ_0 P_0	$\Delta P/P_0$ = 10%		$\Delta P/P_0$ = 5%	
				γ_{0min}	γ_{0max}	γ_{0min}	γ_{0max}
1	10.94	15	1.0(-1) 1.51(-2)	5.3(-2) (-47.0%)	1.76(-1) (76%)	6.4(-2) (-36.0%)	1.38(-1) (38.0%)
		30	3.0(-3) 1.45(-3)	$<10^{-3}$	1.02(-2) (240%)	1.55(-3) (-48.3%)	6.73(-3) (124.3%)
	13.35	15	8.0(-2) 1.29(-2)	4.62(-2) (-42.3%)	1.25(-1) (56.3%)	5.48(-2) (-31.5%)	1.09(-1) (36.3%)
		30	3.0(-3) 4.12(-4)	1.56(-3) (-48%)	4.22(-3) (40.7%)	1.79(-3) (-40.3%)	3.64(-3) (21.3%)
4	10.94	15	1.0 2.6(-2)	0.62 (-38.0%)	1.0	7.79(-1) (-22.1%)	1.0
		30	8.0(-1) 7.27(-3)	$<10^{-3}$	1.0 (25%)	$<10^{-3}$	1.0 (25%)
	13.35	15	1.0 5.82(-3)	0.74 (-26.0%)	1.0	8.53(-1) (-14.7%)	1.0
		30	9.0(-2) 2.0(-4)	4.0(-3) (-95.6%)	1.0 (1011%)	9.9(-3) (-89%)	1.0 (1011%)

*E_b/N_0 for L=1 BER's of 10^{-3} (10.94 dB) and 10^{-5} (13.35 dB)

Note: $a(-n) \equiv a \times 10^{-n}$

TABLE 7-4

SENSITIVITY OF JAMMED BER TO γ_0 FOR THE AGC SQUARE-LAW RECEIVER (M=8)

L	E_b/N_0 (dB)*	E_b/N_J (dB)	γ_0 P_0	$\Delta P/P_0$ = 10%		$\Delta P/P_0$ = 5%	
				γ_{0min}	γ_{0max}	γ_{0min}	γ_{0max}
1	6.97	15	4.71(-2) 8.75(-3)	2.68(-2) (-43.1%)	8.5(-2) (80.5%)	3.18(-2) (-32.5%)	7.04(-2) (49.5%)
		30	1.49(-3) 1.25(-3)	$<10^{-3}$	6.77(-3) (354%)	$<10^{-3}$	3.68(-3) (147%)
	9.09	15	3.77(-2) 7.04(-3)	2.38(-2) (-36.9%)	5.94(-2) (57.6%)	2.78(-2) (-26.3%)	5.17(-2) (37.1%)
		30	1.19(-3) 2.32(-4)	$<10^{-3}$	1.85(-3) (55.5%)	$<10^{-3}$	1.52(-3) (27.7%)
4	6.97	15	1.0 1.65(-2)	2.71(-1) (-72.9%)	1.0	4.48(-1) (-55.2%)	1.0
		30	1.0 8.12(-3)	$<10^{-3}$	1.0	$<10^{-3}$	1.0
	9.09	15	9.72(-1) 1.45(-3)	2.82(-1) (-71%)	1.0 (2.9%)	3.8(-1) (-60.9%)	1.0 (2.9%)
		30	2.35(-2) 2.11(-4)	$<10^{-3}$	1.0 (4155%)	1.07(-3) (-95.4%)	1.0 (4155%)

*E_b/N_0 for L=1 BER's of 10^{-3} (6.97 dB) and 10^{-5} (9.09 dB)

Note: $a(-n) \equiv a \times 10^{-n}$

optimum degradation of the receiver's performance for relatively wide ranges of γ about the optimum value, γ_0. For L=1, on the average the range of acceptable γ, $\gamma_{0max} - \gamma_{0min}$, is about $1.6\gamma_0$ for 10% variation from optimum $P_b(e)$, and about $0.9\gamma_0$ for 5% variation. When L=4, the receiver type and the value of M affect the sensitivities significantly, with acceptable γ ranges generally larger for the clipper receiver, when expressed in terms of γ_0.

In Figures 7-10 (clipper receiver) and 7-11 (AGC receiver) the acceptable ranges of γ_0 based upon 10% performance degradation are depicted for different values of E_b/N_J with L as a parameter (L=1 and 4). In the figures, the nominal γ_0 values are shown with solid lines, and the upper and lower limits are depicted with dotted lines. For the single hop/symbol (L=1) case, the clipper and AGC receiver results give similar ranges of acceptable γ (±40%) for different values of E_b/N_J.

For a higher L (L=4), the clipper receiver results show a symmetric acceptable range of γ for relatively strong jamming, but for weak jamming the upper limit of γ_0 goes to one. On the other hand, the AGC receiver for L=4 is optimally jammed by wideband jamming (γ_0=1) for a greater range of E_b/N_J, and the range of acceptable γ values becomes widely expanded for the whole range of E_b/N_J. In the weak jamming region (high E_b/N_J), we note that it is difficult for the jammer to achieve the desired level of performance degradation and that γ has little effect; this explains the widely expanded acceptable range of γ.

7.2.2 Sensitivity of the Jammed Error Rate to E_b/N_J

For given E_b/N_0, L, M, and receiver type, the worst-case probability of error is achieved by adjusting the partial-band jamming fraction γ to be some value $\gamma_0 = \gamma_0(E_b/N_J)$, where E_b/N_J is estimated from the jammer's projections

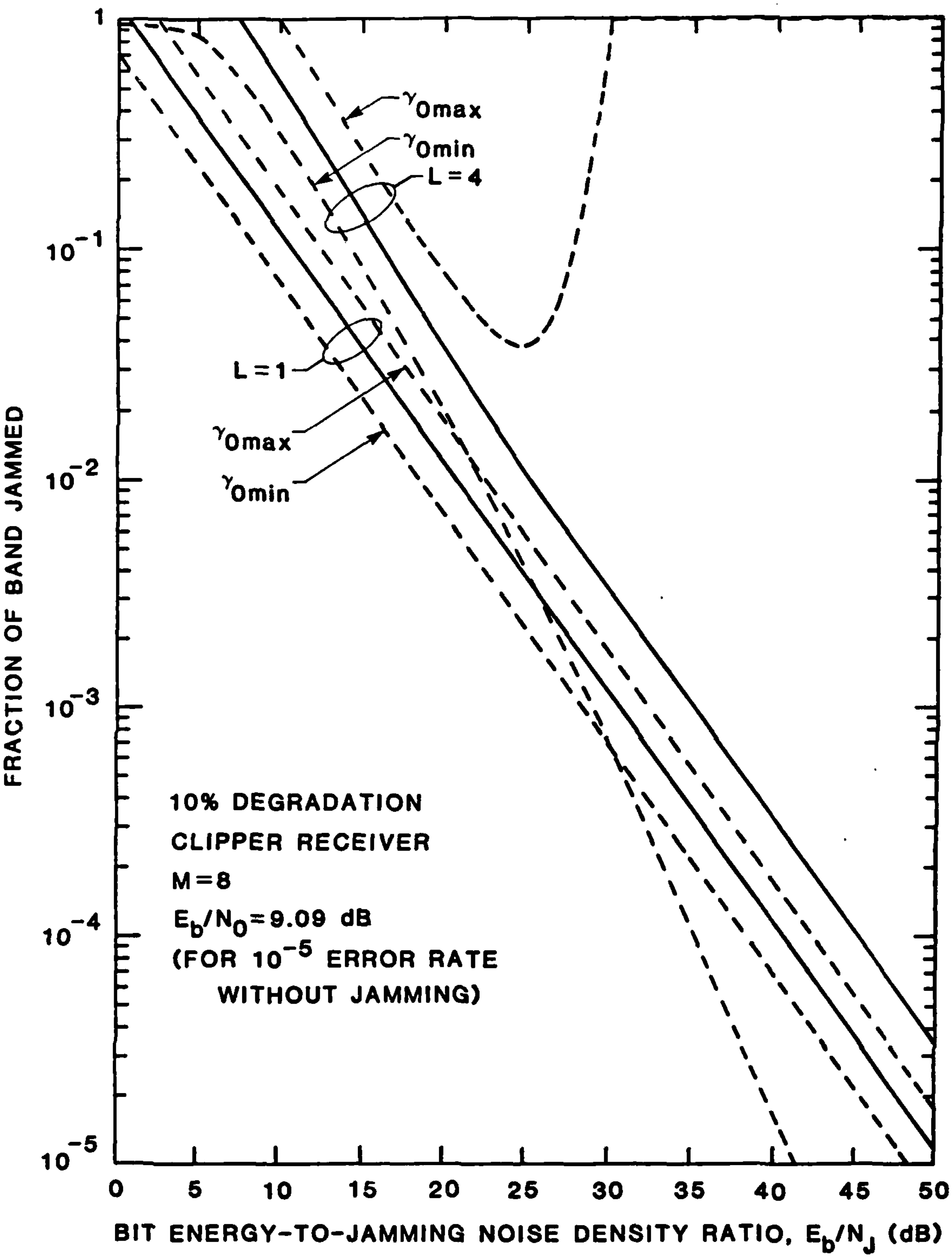


FIGURE 7-10 ACCEPTABLE RANGE OF γ_0 VS. E_b/N_J BASED ON 10% DEGRADATION AGAINST CLIPPER RECEIVER (M=8)

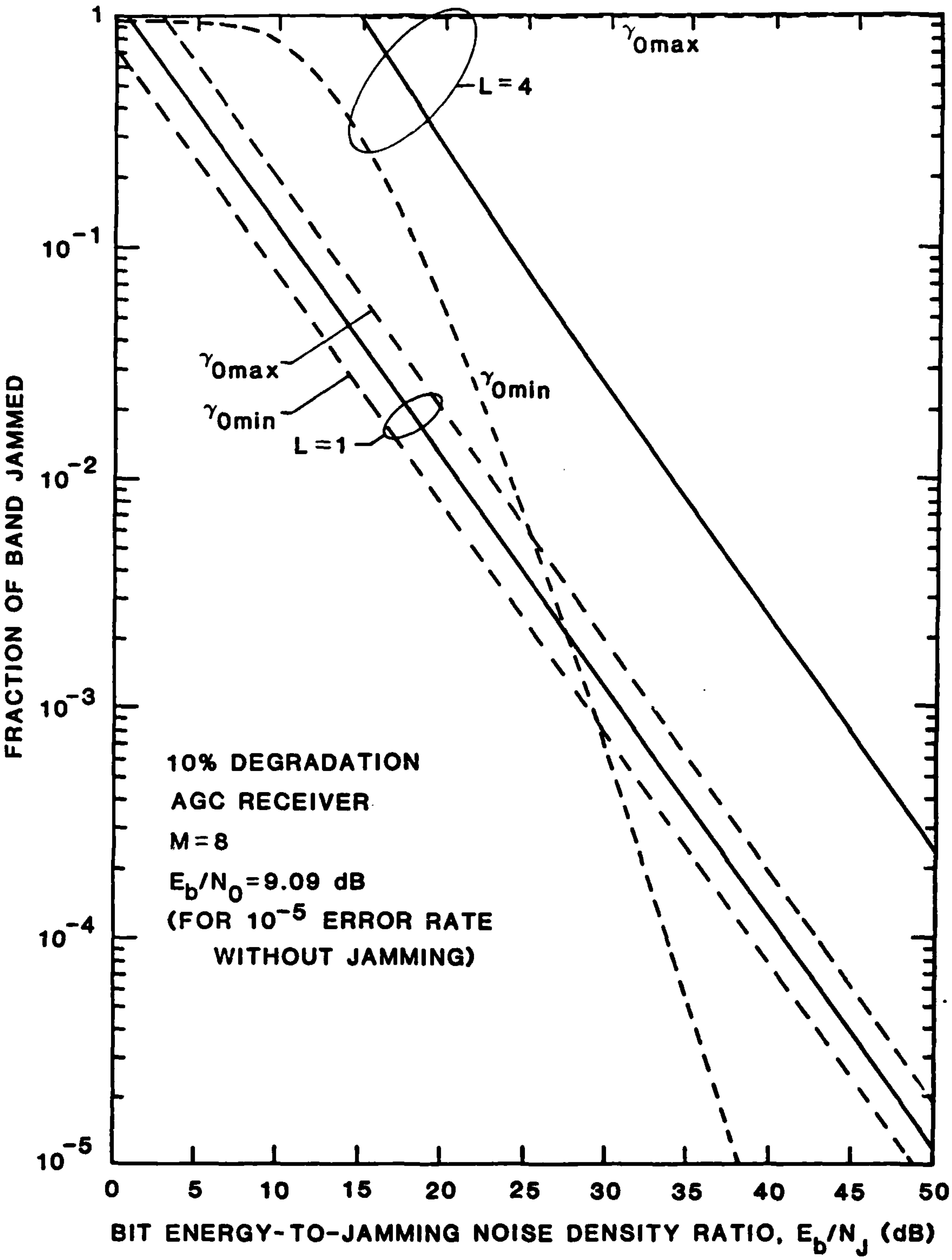


FIGURE 7-11 ACCEPTABLE RANGE OF γ_0 VS. E_b/N_J BASED ON 10% DEGRADATION AGAINST AGC RECEIVER (M=8)

of what the effective signal and noise powers are at the receiver. Assuming that γ_0 has been selected, what is the effect of having used an incorrect value of E_b/N_J?

The dependence of $P_b(e)$ upon E_b/N_J for fixed γ and E_b/N_0 is shown in Figure 7-12 for a typical case (the AGC receiver for L=2, M=8, and $E_b/N_0 = 9.09$ dB). We observe from the figure that if the jammer has set his transmitted power so that $E_b/N_J = \beta_0$, then the error rate of the receiver is P_0 for some particular $\gamma = \gamma_0$. If E_b/N_J actually is less than β_0, then $P_b(e) > P_0$ and the jammer is being more effective, although not as effective as if a different value of γ were used (except, of course, in the region γ_0=1). Therefore in discussing sensitivity we are concerned with the event $E_b/N_J > \beta_0$ for which $P_b(e) = P_0 - \Delta P < P_0$ for a given relative value of $\Delta P/P_0$, and (P_0, β_0) is the point of tangency with the optimum-γ $P_b(e)$ curve.

Tables 7-5 and 7-6 give values of β_{max} for $\Delta P/P_0 = 10\%$ for the clipper and AGC receivers, respectively, when M=8 and L=1 and 4. These data reveal that in general a wider range of E_b/N_J values is acceptable when L=4 than when L=1, for the values of E_b/N_0 selected. For the clipper receiver the tolerance* on E_b/N_J is 0.3 to 0.5 dB for L=1 (7 to 12%), and 0.3 to 1.7 dB (7 to 48%) for L=4. For the AGC receiver, the corresponding tolerances are more difficult to summarize since for higher L, as we noted in Section 4, wideband jamming is optimum; but for L=1, a 0.4 to 0.5 dB (10 to 12%) tolerance in E_b/N_J is acceptable. Figures 7-13 and 7-14 show what we have called β_0 and β_{max} as functions of γ.

*The tolerance is expressed as a percentage difference of β_{max} and β_0 as numeric ratios, not as a percentage difference of decibels.

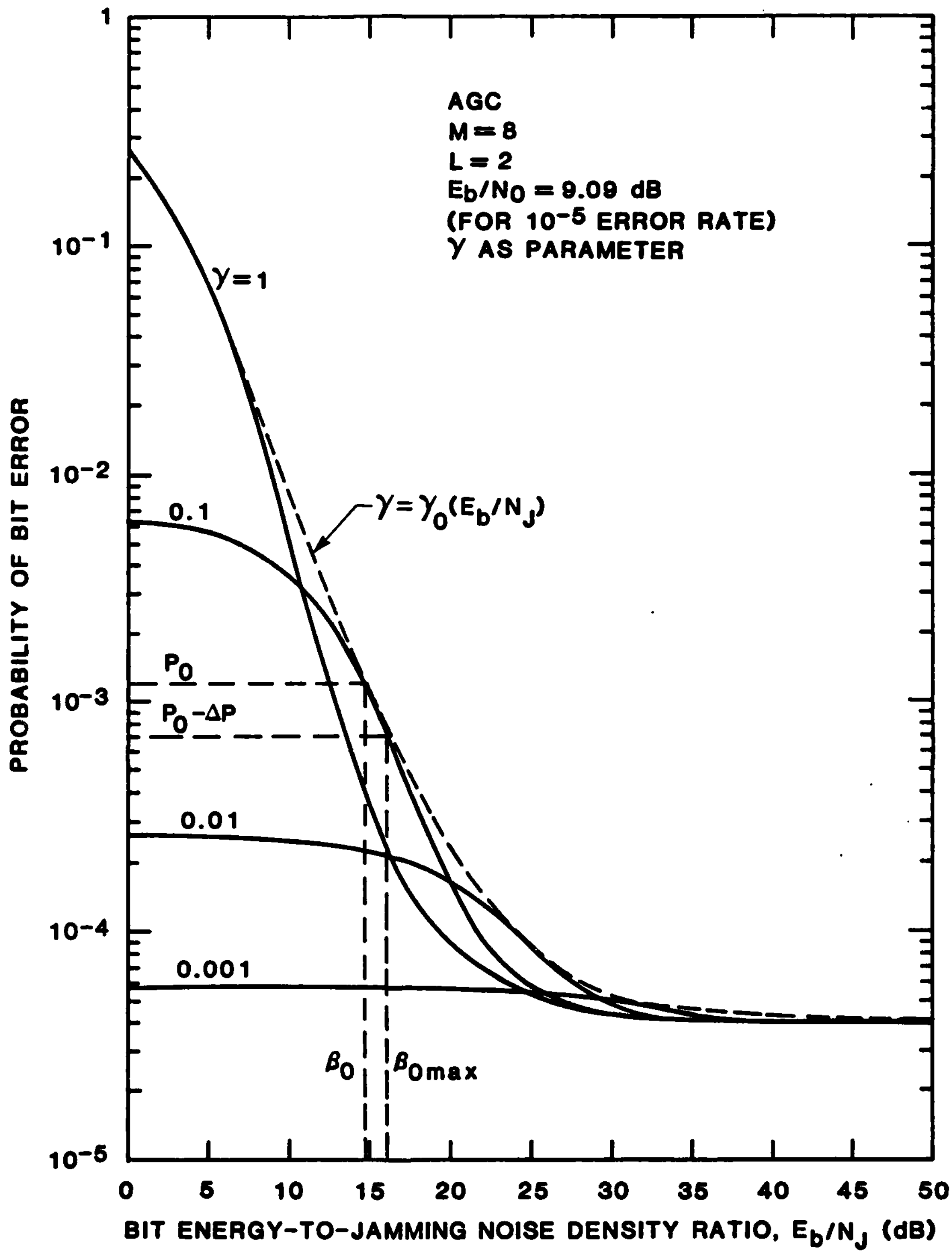


FIGURE 7-12 PROBABILITY OF BIT ERROR VS. BIT ENERGY-TO-JAMMING NOISE DENSITY RATIO FOR FH/MFSK (M=8) AGC RECEIVER WITH L=2 HOPS/SYMBOL WHEN E_b/N_0=9.09 dB SHOWING EFFECTS OF INCORRECT SETTING OF E_b/N_J

TABLE 7-5

SENSITIVITY OF JAMMED BER TO E_b/N_J FOR THE CLIPPER SQUARE-LAW RECEIVER (M=8)

$\Delta P/P_0 = 10\%$					
L	E_b/N_0 (dB)*	γ_0	$\beta_0 = E_b/N_J$ (dB)	$\beta_{max} = E_b/N_{J_{max}}$ (dB)	$\frac{\beta_{max} - \beta_0}{\beta_0}$ (%)
1	6.97 dB	1.0	0	0.5	12.2
		0.1	10.36	10.76	9.6
		0.01	20.0	20.5	12.2
	9.09 dB	1.0	0	0.36	8.64
		0.1	10.0	10.29	6.9
		0.01	20.0	20.3	7.2
4	6.97 dB	1.0	8.7	9.1	9.6
		0.1	19.29	21.0	48.3
		0.01	25.71	**	-
	9.09 dB	1.0	7.6	7.9	7.2
		0.1	16.43	16.86	10.4
		0.01	25.0	26.0	25.9

*E_b/N_0 for L=1 BER's of 10^{-3} (6.97 dB) and 10^{-5} (9.09 dB)

**$\Delta P/P_0$ is always < 10%

TABLE 7-6

SENSITIVITY OF JAMMED BER TO E_b/N_J FOR

THE AGC SQUARE-LAW RECEIVER (M=8)

$\Delta P/P_0 = 10\%$					
L	E_b/N_0 (dB)*	γ_0	$\beta_0 = E_b/N_J$ (dB)	$\beta_{max} = E_b/N_{J_{max}}$ (dB)	$\frac{\beta_{max} - \beta_0}{\beta_0}$ (%)
1	6.97 dB	1.0	0	0.5	12.2
		0.1	12.14	12.57	10.4
		0.01	22.14	22.59	10.9
	9.09 dB	1.0	0	0.43	10.4
		0.1	10.21	10.63	10.2
		0.01	20.71	21.16	10.9
4	6.97 dB	1.0	26.7	**	-
		0.1	33.5	**	-
		0.01	38.4	**	-
	9.09 dB	1.0	14.6	14.9	7.2
		0.1	23.57	25.35	50.7
		0.01	31.0	**	

* E_b/N_0 for L=1 BER's of 10^{-3} (6.97 dB) and 10^{-5} (9.09 dB).

** $\Delta P/P_0$ is always < 10%

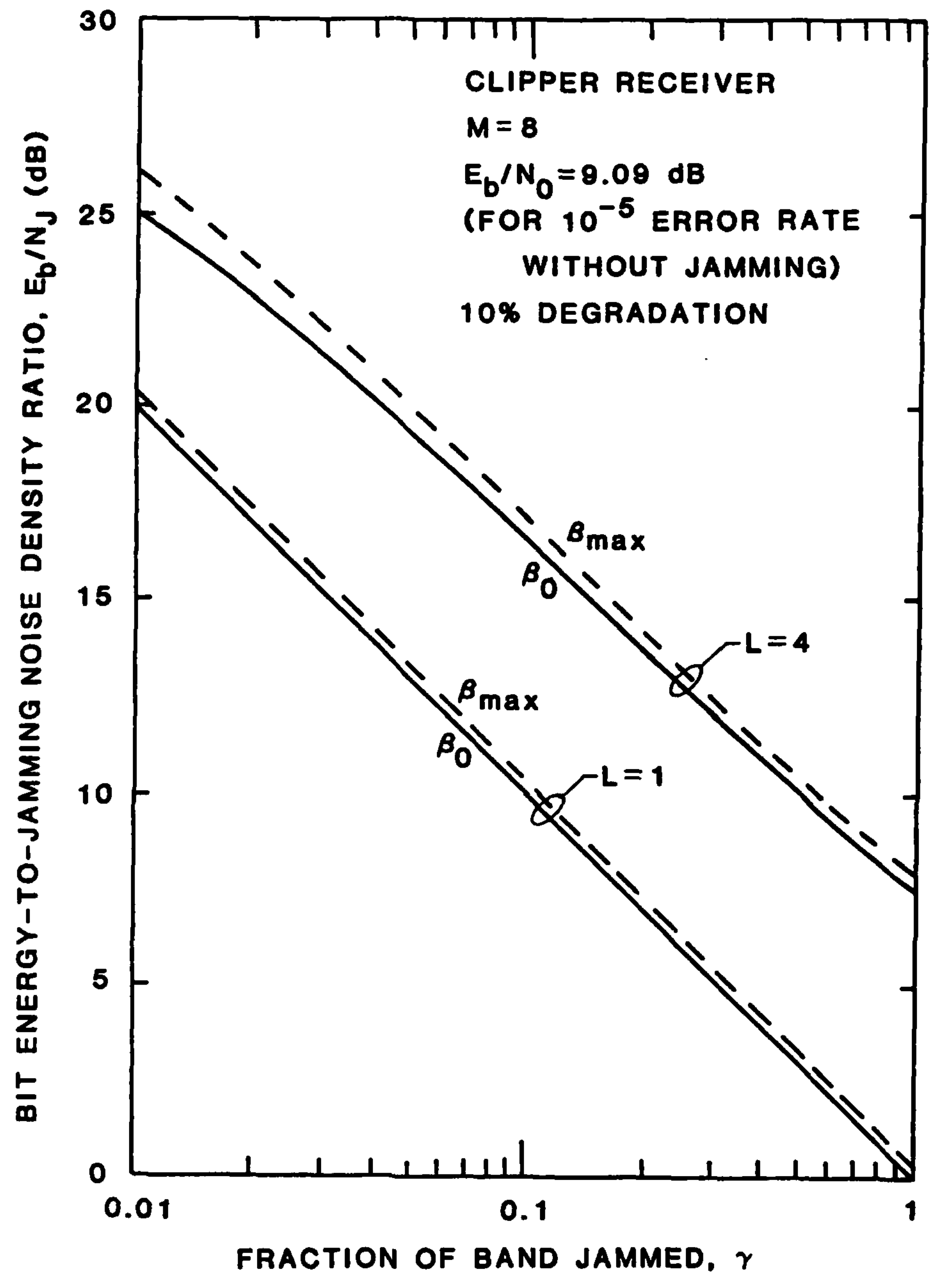


FIGURE 7-13 ALLOWABLE RANGE OF BIT ENERGY-TO-JAMMING NOISE DENSITY RATIO FOR 10% DEGRADATION AS A FUNCTION OF JAMMING FRACTION AGAINST CLIPPER RECEIVER FOR MFSK/FH (M=8) WHEN E_b/N_0=9.09 dB

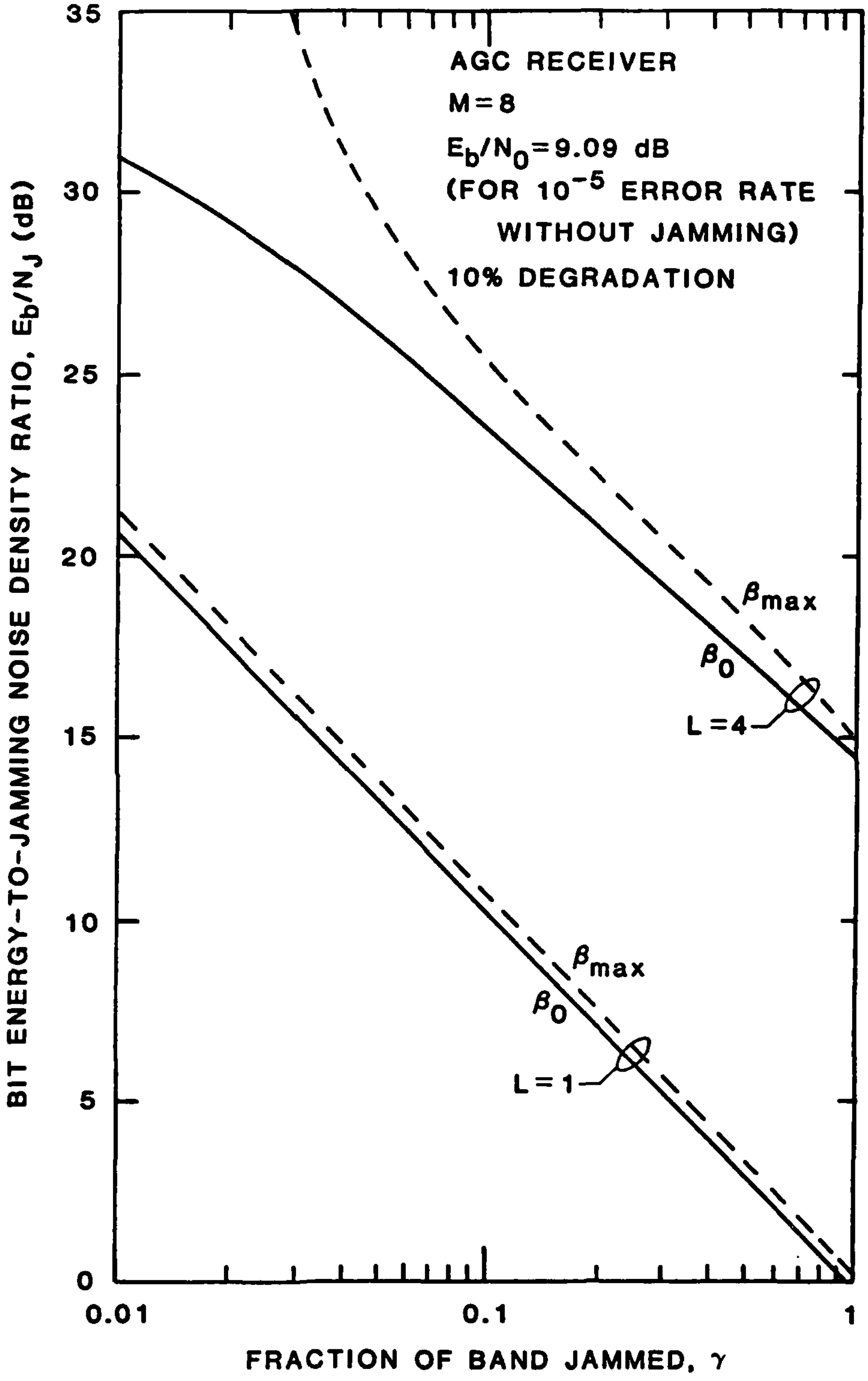


FIGURE 7-14 ALLOWABLE RANGE OF BIT ENERGY-TO-JAMMING NOISE DENSITY RATIO FOR 10% DEGRADATION AS A FUNCTION OF JAMMING FRACTION AGAINST AGC RECEIVER FOR MFSK/FH (M=8) WHEN E_b/N_0=9.09 dB

A lower bound on β_{max} can be derived for the AGC receiver. For no thermal noise ($E_b/N_0 \to \infty$),

$$P_0 \to k/\beta_0^L \equiv P_{0,\infty}; \qquad (7\text{-}9)$$

then from this limiting case, since $P_0 > P_{0,\infty}$,

$$\frac{\Delta\beta}{\beta_0} > \frac{1}{L}\ \frac{\Delta P}{P_0} \qquad (7\text{-}10)$$

where $\Delta\beta = \beta_{max} - \beta_0$.

In conclusion we may say that although the tolerances on E_b/N_J are smaller than those on γ for acceptable jamming results, they are both of an order of magnitude which is realizable in practice.

7.3 CONCEPTUAL JAMMER SYSTEM DESIGN

Based on our analyses of the effects of optimum partial-band noise jamming and on basic scenario-dependent and receiver-dependent jamming system considerations, we now discuss the conceptual design of a real-time jammer system. A block diagram of the elements of such a system is given in Figure 7-15.

In order to implement the jamming which analysis shows to be optimum, certain parameters are required as inputs to the system. These parameters can be categorized as either a priori or measured. Among the parameters which are necessarily a priori because they cannot be measured in real time are L, M, and N_0; these are assumed to be known from intelligence. Measured parameters include the system bandwidth, the hop rate, and the signal and jammer powers at the receiver. The relative locations of the receiver and the communications transmitter, together with knowledge of radiated power, antennas, and propagation factors, are needed to determine the powers accurately. It has been shown that

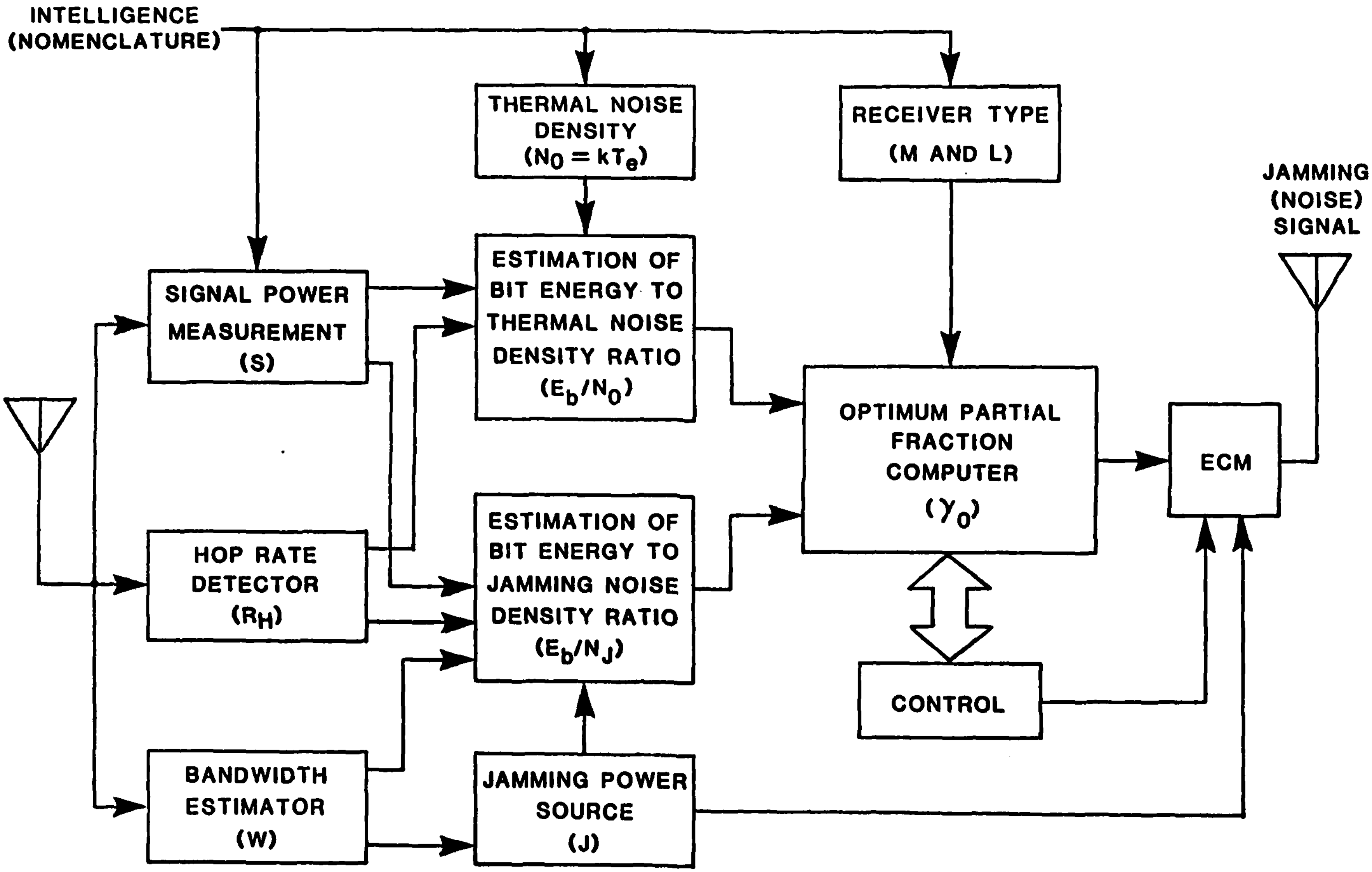


FIGURE 7-15 CONCEPTUAL DESIGN OF A REAL-TIME JAMMER SYSTEM

effective jamming can be performed with less than ideal parameters values, so we assume that implementation of these measurements is feasible.

With the measured hop rate R_H and the signal power S, we can estimate the signal energy per bit E_b using

$$E_b = \frac{S}{R_b} = \frac{S}{R_H \frac{\log_2 M}{L}} , \tag{7-11}$$

where R_b is the bit rate, M is the alphabet size and L is the number of hops/symbol. The system thermal noise density N_0 is also assumed to be available to a jammer through intelligence. With knowledge of the system bandwidth W and the jamming power J, the effective jamming noise power density N_J is expressed by $N_J = J/W$.

Using the values for E_b, N_0, and N_J, we are able to compute the bit energy-to-thermal noise density ratio (E_b/N_0), and the bit energy-to-jamming noise density ratio (E_b/N_J) upon which the optimum jamming fraction depends. The optimum jammer design basically utilizes knowledge of the processing scheme (or receiver type, M, and L), E_b/N_0, and E_b/N_J to determine the optimum jamming fraction γ_0 that the jammer can use with his limited jamming power to victimize effectively the communication link.

The jammer design of Figure 7-15 assumes that matrices of γ_0 values are stored in memory. With knowledge of the required independent parameters (receiver type, M, L, E_b/N_0, and E_b/N_J), the jammer may automatically look up the value of γ_0 from the memory. Figure 7-16 shows a flow diagram for such a jammer control algorithm which can be used against the clipper and AGC receivers. Considering possible limitations on memory size, it seems reasonable to choose increments, for example, of 5 dB for tabulating γ_0 values as a function of E_b/N_J and using interpolation and extrapolation routines when required.

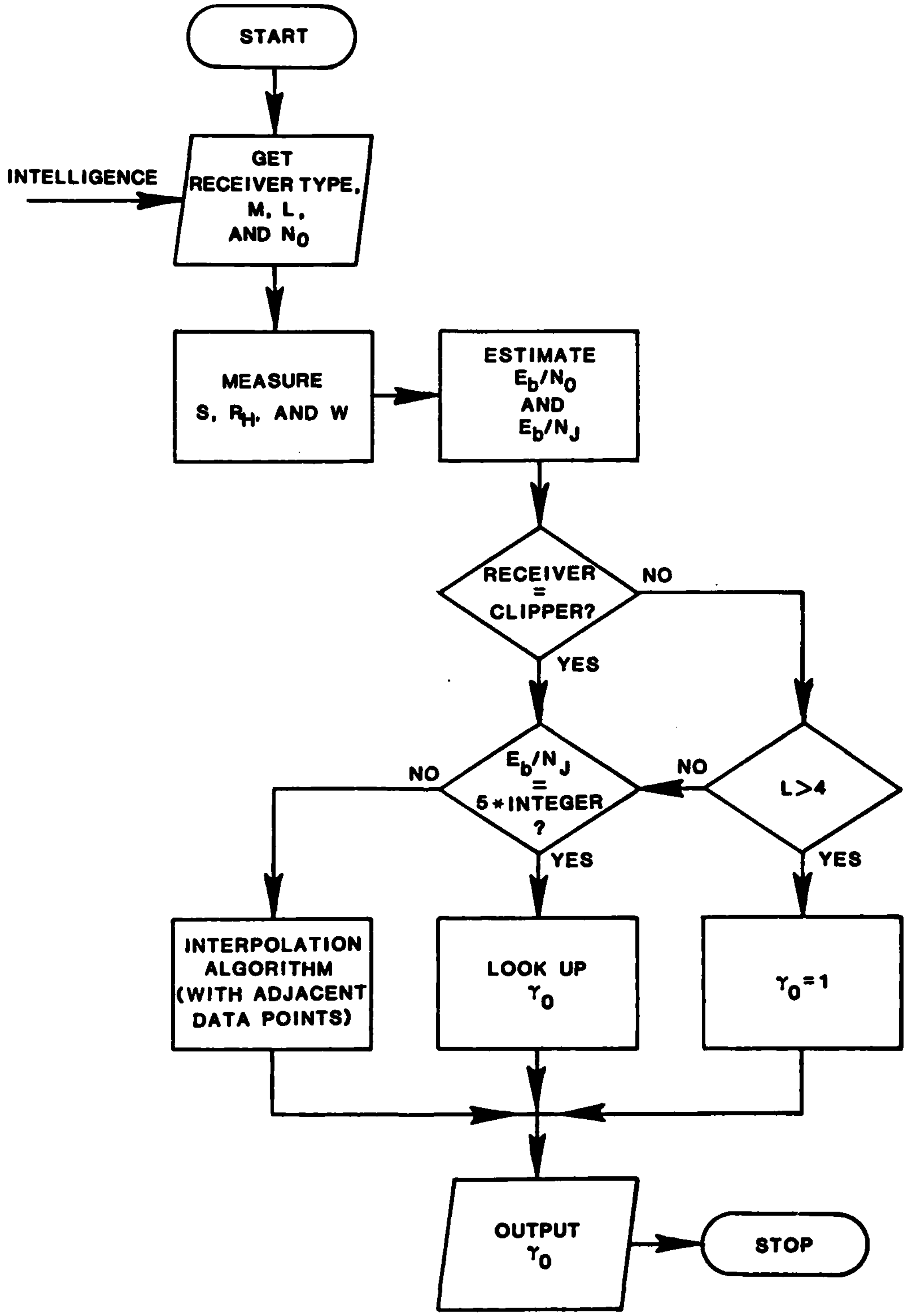


FIGURE 7-16 FLOW DIAGRAM OF OPTIMUM UNIVERSAL JAMMING STRATEGY AGAINST L-HOPS/SYMBOL FH/MFSK COMMUNICATION SYSTEMS

The amount of read-only memory (ROM) required for this look-up table can be computed using the number of values for each input parameter shown in Table 7-7. We assume that the values of γ_0 are stored as unsigned binary floating-point numbers with an 8-bit exponent and a 24-bit fraction; this is equivalent to 6 to 7 decimal digit accuracy. Thus each number requires 32 bits of storage. From Table 7-7 we find that the total storage, in Kbits (1K = 1024), needed for the look-up table is $\lceil 32 \times 3 \times 5 \times 6 \times 5 \times 11/1024 \rceil =$ 155 Kbits of ROM. If the data are organized as four 8-bit bytes at consecutive memory addresses for each floating point number, then a 20K x 8 bit ROM would suffice. As a practical design, this would be rounded up to a standard size of 32K x 8 bits. This amount of memory is available as mask-programmed ROM in a single 28-pin integrated circuit specified over the full military temperature range with access times as low as 250 ns [20, p. 3787]. One-chip PROM with this capacity is available [20, p. 3767] but only in commercial temperature-range devices. However two 16K x 8 bit PROMs, which are available in full military temperature-range devices [20, p. 3767], would suffice. We conclude that there is no difficulty in storing the look-up table for γ_0; indeed, finer resolution in some parameters could even be accommodated with only a small increase in the number of components in the hardware.

TABLE 7-7

PARAMETERS FOR JAMMER MEMORY REQUIREMENTS

PARAMETER	NUMBER OF VALUES
Receiver Type	3
M	5
L	6
E_b/N_0 (dB)	5
E_b/N_J (dB)	11

8.0 ANALYSIS OF FH/MFSK ERROR RATE UNDER TONE JAMMING

In the previous sections we have been concerned with the effects of partial-band noise jamming on FH/MFSK systems with multiple hops per symbol. We now consider the situation in which the jammer chooses to employ tones or sinusoids as his interfering signal. The term partial-band tone jamming may be applied to the jamming wave form in the sense that a number of tones (q) are placed relatively close to one another in some fraction of the hopping system bandwidth (W).

The modelling and analysis of tone jamming differ from that of noise jamming in several respects. Since the spectrum of the jamming waveform using tones is discrete, the fraction of the band thus jammed is not generally equal to the fraction of hopping slots jammed, $\gamma \triangleq q/N$, where $N = W/B$ is the number of hopping slots in the system bandwidth. For noise jamming we assumed that all or none of the M symbol frequency slots were jammed because of an unbroken spectral distribution of noise power in γW Hz, and neglecting edge effects for hops that fall only partially into the jammed portion of the band. For tone jamming, except for the special case of adjacent tones, we must consider jamming events in which some of the M slots are jammed while others are not. Another difference between noise and tone jamming is the signal-like character of jamming tones and the possibility of phase cancellation of the communications signal. This possibility requires consideration of the relative phase difference between the jamming tone and the signal in our analysis, in addition to their respective powers.

In what follows we first formulate the bit error probability for the FH/MFSK receiver for a general model of tone jamming, then consider specific tone jamming models and give numerical results for the error proba-

bilities for various parametric situations. For convenience we have selected the conventional square-law linear combining FH/MFSK receiver shown in Figure 8-1 for our analysis and calculations.

8.1 PROBABILITY OF ERROR UNDER TONE JAMMING

This section first states the assumptions used in our analysis of the jamming and introduces the notation we have adopted. We then describe the possible jamming events and derive the conditional probability of symbol error for an arbitrary jamming event. Finally, the total probability of error is expressed as the average of the conditonal probability of error over all jamming events.

8.1.1 General Tone Jamming Model

The FH/MFSK signal is assumed to be randomly hopped within a system bandwidth $W = NB$, where $B = R_H = 1/\tau$ and R_H is the hopping rate. The M symbol frequencies are assumed to be spaced B Hz apart, so the M-ary signal occupies one of M contiguous slots on each hop and there are N-M+1 possible hopping positions for the symbol. For L hops per symbol ($L = 1, 2, \ldots$) there are LM opportunities for the symbol to include a slot occupied by a jamming tone.

The jammer is assumed to share J watts of power equally among q tones ($q = 1, 2, \ldots, N$), each of which is centered exactly in one of the N available slots. Let the L hops for a given symbol be referred to individually by the index k ($k = 1, 2, \ldots, L$). Then the jamming events for the kth hop can be described in terms of which of the M symbol frequencies are jammed, and which are not. In general there are 2^M possibilities for a given hop, which we may specify by the indicator vector

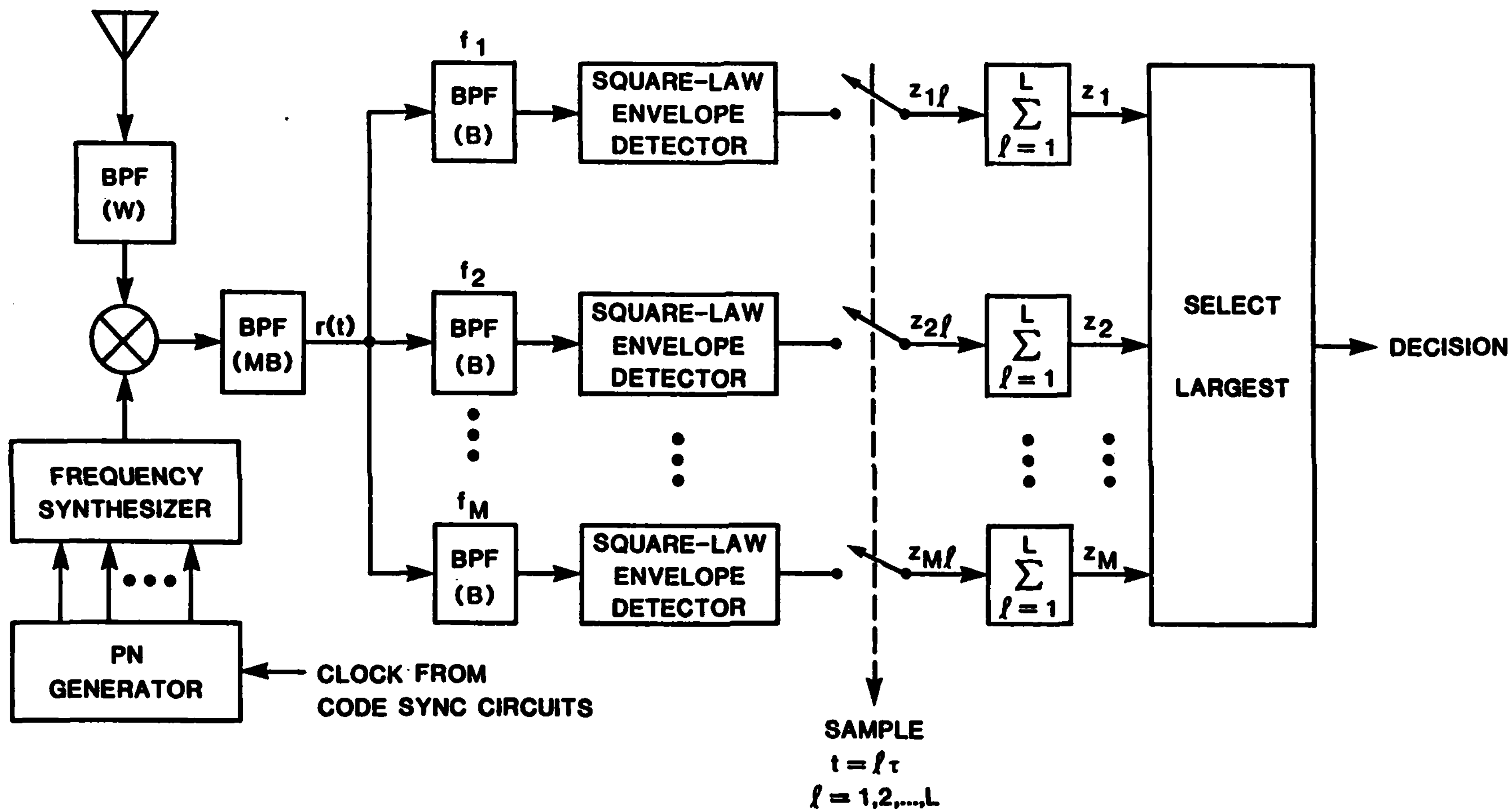


FIGURE 8-1 RECEIVER FOR MFSK/FH WITH FAST HOPPING

$$\underline{\nu}_k = (\nu_{1k}, \nu_{2k}, \ldots, \nu_{Mk}) \tag{8-1a}$$

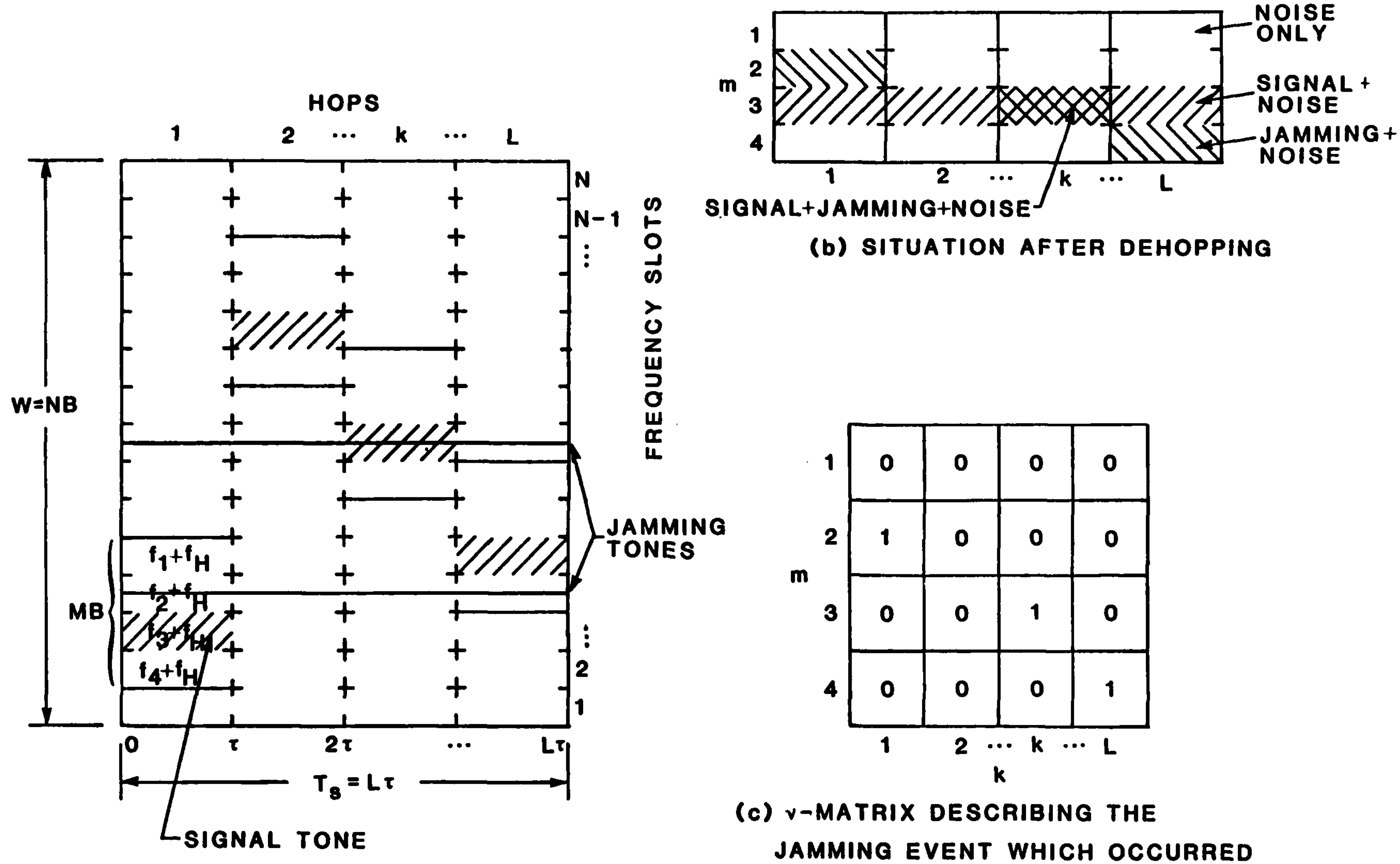
where

$$\nu_{mk} = \begin{cases} 1 \text{ if symbol slot m is jammed on hop k} \\ 0 \text{ if not;} \end{cases} \tag{8-1b}$$
$$m = 1, 2, \ldots, M; \; k = 1, 2, \ldots, L.$$

Thus for the L hops together there are 2^{ML} possible jamming events, and these can be specified individually by the $M \times L$ indicator matrix $\boldsymbol{\nu} = [\nu_{mk}]$. Figure 8-2 gives an example jamming event pattern and the matrix corresponding to it. Part (a) of the figure shows an FH/MFSK signal with M = 4 hopping within the N frequency slots L times for one symbol. In this example, the information is shown to be conveyed by selection of the baseband frequency f_3, and two jamming tones are postulated. The second hop is not jammed, while the first, kth, and Lth hops are shown jammed because one of the symbol's four slots has hopped into positions containing a jamming tone. After dehopping, the situation is as shown in Figure 8-2(b). Certain time-frequency slots contain noise only, certain ones contain the signal plus noise; certain ones contain jamming plus noise; and certain ones contain jamming, signal, and noise. The use of the ν_{mk} notation to describe what jamming event has occurred is illustrated in part (c) of the figure; for the example, $\nu_{21} = \nu_{3k} = \nu_{4L} = 1$, while all the other ν's are zero.

Using the ν_{mk} notation, and assuming without loss of generality that the signal frequency is $f = f_1$, the square-law envelope detector samples z_{mk} for the receiver of Figure 8-1 are

$$\begin{aligned} z_{1k} &= \left(\sqrt{2S}\cos\theta_k + \nu_{1k}\sqrt{2J_0}\cos\phi_{1k} + n_{c1k}\right)^2 \\ &\quad + \left(\sqrt{2S}\sin\theta_k + \nu_{1k}\sqrt{2J_0}\sin\phi_{1k} + n_{s1k}\right)^2, \quad m = 1 \end{aligned} \tag{8-2a}$$



(a) SITUATION BEFORE DEHOPPING

(b) SITUATION AFTER DEHOPPING

m \ k	1	2 ···	k ···	L
1	0	0	0	0
2	1	0	0	0
3	0	0	1	0
4	0	0	0	1

(c) ν-MATRIX DESCRIBING THE JAMMING EVENT WHICH OCCURRED

FIGURE 8-2 EXAMPLE JAMMING EVENT

and

$$z_{mk} = \left(\nu_{mk}\sqrt{2J_0}\cos\phi_{mk} + n_{cmk}\right)^2 + \left(\nu_{mk}\sqrt{2J_0}\sin\phi_{mk} + n_{smk}\right)^2, \quad m = 2, 3, \ldots, M; \quad k = 1, 2, \ldots, L, \tag{8-2b}$$

where S is the signal power, $J_0 \equiv J/q$ is the jamming power per tone, θ is the signal phase, ϕ is the jamming tone phase, and n_c and n_s are quadrature components of the thermal noise. The 2ML quadrature noise components are independent and each is a zero-mean Gaussian random veriable with variance σ_N^2.

8.1.2 Conditional Error Probabilities

For a given jamming event, described by the matrix $\mathbf{v}$, the square-law envelope detector samples z_{mk} are modeled as in (8-2). As indicated in Figure 8-1, the symbol decision is based on selecting the largest of the decision variables z_m, m = 1, 2,..., M, where

$$z_m = \sum_{k=1}^{L} z_{mk}. \tag{8-3}$$

We observe that each z_{mk} is $\sigma_N^2 = N_0B$ times a noncentral chi-squared random variable with two degrees of freedom and noncentrality parameter λ_{mk}, where

$$\lambda_{1k} = \frac{1}{\sigma_N^2}\left[2S + 2\nu_{1k} J_0 + 4\nu_{1k}\sqrt{SJ_0}\cos(\theta_k - \phi_{1k})\right] \tag{8-4a}$$

and

$$\lambda_{mk} = 2\nu_{mk} J_0/\sigma_N^2, \quad m \geq 2. \tag{8-4b}$$

In writing (8-4), we have used the fact that $\nu_{mk}^2 = \nu_{mk}$. Since the signal and jammer phases in (8-4a) are unknown, we must regard z_{1k} as conditionally chi-squared, with random parameter λ_{1k}.

The decision variables, being sums of uniformly scaled chi-squared random variables, are also σ_N^2 times noncentral chi-squared variables, with 2L degrees of freedom and noncentrality parameters

$$\lambda_1 = \frac{2}{\sigma_N^2}\left[LS + \ell_1 J_0 + 2\sqrt{SJ_0}\sum_{k=1}^{L}\nu_{1k}\cos\left(\theta_k - \phi_{1k}\right)\right] \qquad (8\text{-}5a)$$

and

$$\lambda_m = 2\ell_m J_0/\sigma_N^2, \; m \geq 2; \qquad (8\text{-}5b)$$

where

$$\ell_m \triangleq \sum_{k=1}^{L}\nu_{mk}, \; m = 1, 2, \ldots, M. \qquad (8\text{-}6)$$

The quantity ℓ_m defined in (8-6) is the number of times in L hops that symbol frequency slot m is jammed. The noncentrality parameter λ_1 for the signal channel given by (8-5a) may also be written as

$$\lambda_1 = \frac{2}{\sigma_N^2}\left[LS + \ell_1 J_0 + 2\sqrt{SJ_0}\,\zeta(\ell_1)\right] \qquad (8\text{-}7)$$

where the random variable $\zeta(\ell_1)$ is the sum of ℓ_1 cosines of signal-jammer phase differences. If $\ell_1=0$, we define $\zeta(0)=0$. Thus the error probability for a given jamming event depends only on the ℓ_m, $m = 1,2,\ldots, M$, or numbers of hops jammed in each symbol frequency slot, rather than on the specific pattern ν.

Given the set of ℓ_m values $\underline{\ell} = (\ell_1, \ell_2, \ldots, \ell_M)$, the probability of a symbol decision error is given by

$$P_s(e|\underline{\ell}) = 1 - \Pr\{z_1 > z_2, z_1 > z_3, \ldots, z_1 > z_M|\underline{\ell}\}$$

$$= 1 - \int_0^\infty d\alpha \; p_{z_1}(\alpha|\ell_1) \prod_{m=2}^{M} \int_0^\alpha d\beta_m \, p_{z_m}\left(\beta_m|\ell_m\right) \qquad (8\text{-}8)$$

where $p_{z_m}(\cdot)$ is the probability density function for z_m. For the nonsignal slots,

$$p_{z_m}\left(\beta_m|\ell_m\right) = \frac{1}{2\sigma_N^2} \exp\left[-\frac{1}{2}\left(\frac{\beta_m}{\sigma_N^2} + \lambda_m\right)\right]\left(\frac{\beta_m}{\sigma_N^2\,\lambda_m}\right)^{(L-1)/2}$$

$$\cdot\, I_{L-1}\left(\sqrt{\lambda_m\beta_m/\sigma_N^2}\right), \quad m = 2,\ldots, M. \qquad (8\text{-}9)$$

Their distribution functions are

$$F_{z_m}(\alpha) = \int_0^\alpha d\beta \; p_{z_m}(\beta) = \Pr\{z_m < \alpha\}$$

$$= P_{\chi^2}\left(\frac{\alpha}{2\sigma_N^2};\; 2L, \lambda_m\right) \qquad (8\text{-}10)$$

where $P_{\chi^2}(\cdot\ ;\ \mu,\ \lambda)$ is the distribution function for a noncentral chi-squared variable with μ degrees of freedom and noncentrality parament λ. Alternately, we can use the expression

$$F_{z_m}(\alpha) = 1 - Q_L(\sqrt{\lambda_m},\ \sqrt{\alpha/\sigma_N}), \tag{8-11}$$

where $Q_L(\cdot,\cdot)$ is the generalized Q-function [25].

The pdf for the signal channel decision variable z_1 is

$$\begin{aligned} p_{z_1}(\alpha|\ell_1) &= E_\zeta\left[p_{z_1}(\alpha|\ell_1,\ \zeta)\right] \\ &= \int_{-\ell_1}^{\ell_1} dx\ p_{z_1}(\alpha|\ell_1,x)\ p_\zeta(x|\ell_1) \end{aligned} \tag{8-12}$$

where the conditional pdf $p_{z_1}(\alpha|\ell_1,\ x)$ is the chi-squared form given by (8-9). The random variable ζ, being the sum of ℓ_1 cosines of real arguments, is bounded by $\pm\ell_1$; hence the integral in (8-12) is taken over this range.

For $\ell_1 = 1$, it can be shown [23, p. 133] that the pdf of $\zeta(1)$ is

$$p_\zeta(x|1) = \frac{1}{\pi\sqrt{1 - x^2}}, \quad |x| < 1, \tag{8-13}$$

with characteristic function

$$\phi_\zeta(j\mu|1) = \int_{-1}^{1} dx\ e^{j\mu x}\ p_\zeta(x|1) = J_0(\mu). \tag{8-14}$$

Thus for $\ell_1 > 1$, in principle we can find the pdf of $\zeta(\ell_1)$ by taking the inverse transform of the ℓ_1th power of $J_0(\mu)$:

$$p_\zeta(x|\ell_1) = \frac{1}{2\pi}\int_{-\infty}^{\infty} d\mu \; e^{-j\mu x}\,[J_0(\mu)]^{\ell_1}$$

$$= \frac{1}{\pi}\int_0^{\infty} d\mu \; \cos\mu x \,[J_0(\mu)]^{\ell_1} \tag{8-15}$$

where the second form follows from Euler's formula and the fact that $J_0(\cdot)$ and the cosine are even functions and the sine is an odd function. The calculation of this density function has been investigated by Slack [24]. For the case of $\ell_1 = 2$, (8-15) can be expressed in closed form as

$$p_\zeta(x|\ell_1 = 2) = \frac{1}{\pi^2} K\left(\sqrt{1 - \frac{x^2}{4}}\right) \tag{8-16}$$

where $K(k)$ is the complete elliptic integral of the first kind with modulus k which may also be expressed in terms of the Gauss hypergeometric function by the relation [4, eq. 17.2.19 and 17.3.9]

$$K(k) = \frac{\pi}{2}\; {}_2F_1(\tfrac{1}{2}, \tfrac{1}{2}; 1; k^2). \tag{8-17}$$

However, for $\ell_1 > 2$, an exact closed form is not available, and $p_\zeta(x|\ell_1)$ is best obtained by numerical computation of (8-15) or repeated numerical convolution of (8-13) with itself.

---

* This result can be obtained from (8-15) by using an integral representation of $J_0^2(x)$, as was done in [24], or more directly from the self-convolution of (8-13) using [2, eq. 3.152.10] to evaluate the convolution integral.

Using (8-11) and (8-12), the conditional probability of symbol error becomes

$$P_s(e|\underline{\ell}) = 1 - \int_0^\infty d\alpha \int_{-\ell_1}^{\ell_1} dx\; p_{z_1}(\alpha|\ell_1, x)\; p_\zeta(x|\ell_1) \cdot \prod_{m=2}^{M} \left[1 - Q_L(\sqrt{\lambda_m}, \sqrt{\alpha/\sigma_N^2})\right] \quad \text{(8-18a)}$$

with λ_m given by (8-5b). We can write an alternate form, which is more suitable for numeric computation, by noting that the result of the inner integral in (8-18a) is itself a density function, and therefore integrates to 1. This allows us to take the subtraction operation inside the outer integral and write

$$P_s(e|\underline{\ell}) = \int_0^\infty d\alpha \int_{-\ell_1}^{\ell_1} dx\; p_{z_1}(\alpha|\ell_1, x)\; p_\zeta(x|\ell_1) \left\{1 - \prod_{m=2}^{M}\left[1 - Q_L(\sqrt{\lambda_m}, \sqrt{\alpha/\sigma_N^2})\right]\right\}. \quad \text{(8-18b)}$$

The form (8-18b) is more suitable for numeric computations because the onus of computing many significant digits is removed from the entire double integral and placed on only the generalized Q functions, for which a reasonably efficient numerical algorithm is available [25]. The final transformation needed for actual numerical computations is achieved by making the substitution $y = \alpha/\sigma_N^2$ in (8-18b) to obtain the form

$$P_s(e|\underline{\ell}) = \int_0^\infty dy \int_{-\ell_1}^{\ell_1} dx\; \sigma_N^2\, p_{z_1}(\sigma_N^2\, y|\ell_1, x)\; p_\zeta(x|\ell_1) \left\{1 - \prod_{m=2}^{M}\left[1 - Q_L(\sqrt{\lambda_m}, \sqrt{y})\right]\right\}. \quad \text{(8-18c)}$$

Referring to (8-9) for the form of $p_{z_1}(\cdot)$, we see that the form (8-18c) removes all dependence on the noise variance σ_N^2 except for that embedded in the signal-to-noise and jamming-to-noise ratios.

A further computational savings can be realized by taking advantage of the fact that the forms given by (8-18) are invariant under a permutation of the non-signal channel jamming events. If we re-order the parameters λ_k, k = 2, 3,..., M, the result is unchanged. The physical interpretation of this is that the error probability depends only on the fact that a non-signal channel has a certain number of hops jammed, and not upon which specific non-signal channel it is. To put it another way, all non-signal channels are identical in their mechanism for influencing the decision.

8.1.3 Total Error Probability

In terms of the error probabilities conditioned on jamming events, the total symbol error probability is written

$$P_s(e) = \sum_{\underline{\ell}} \pi_L(\underline{\ell})\, P_s(e|\underline{\ell}) \tag{8-19}$$

where $P_s(e|\underline{\ell})$ is given by (8-18) and the jamming event probabilities $\pi_L(\underline{\ell})$ are given by

$$\pi_L(\ell_1, \ell_2,\ldots, \ell_M) \triangleq \Pr\left\{ \sum_{k=1}^{L} \nu_{1k} = \ell_1, \sum_{k=1}^{L} \nu_{2k} = \ell_2,\ldots, \sum_{k=1}^{L} \nu_{mk} = \ell_M \right\}$$

$$= \Pr\left\{ \sum_{k=1}^{L} \underline{\nu}_k = \underline{\ell} \right\}. \tag{8-20}$$

The subscript L in the notation Π_L emphasizes that these are jamming event probabilities over L hops per symbol. From (8-18) and (8-5), we observe that the symbol error probability depends directly upon several parameters:

$$P_s(e) = P_s(e;\ S/\sigma_N^2\ ,\ J_0/\sigma_N^2\ ,\ L,\ M). \tag{8-21}$$

For $M = 2^K$ we may identify the appropriate energy and bit-related parameters as

$$\frac{E_b}{N_0} = \frac{1}{K}\cdot\frac{E_s}{N_0} = \frac{L}{K}\cdot\frac{E_h}{N_0} = \frac{L}{K}\cdot\frac{S}{\sigma_N^2} \tag{8-22a}$$

$$\frac{E_b}{N_J} = \frac{L}{K}\cdot\frac{E_h}{N_J} = \frac{L}{K}\cdot\frac{S}{J/N} = \frac{L}{K}\cdot\frac{N}{q}\cdot\frac{S/\sigma_N^2}{J_0/\sigma_N^2}\ . \tag{8-22b}$$

Thus the bit error probability is found from

$$P_b(e) = P_b\left(e;\ \gamma \equiv \frac{q}{N},\ \frac{E_b}{N_0},\ \frac{E_b}{N_J},\ L,\ M\right).$$

$$= \frac{M/2}{M-1}\ P_s\left(e;\ \frac{K}{L}\cdot\frac{E_b}{N_0},\ \frac{1}{\gamma}\cdot\frac{E_b/N_0}{E_b/N_J}\ ,\ L,\ M\right). \tag{8-23}$$

In our numerical studies which follow, we are interested in how $P_b(e)$ varies as a function of E_b/N_J, with the other parameters held constant, under several tone jamming models. Our attention is particularly directed at determining whether the error rate under worst case tone jamming (the maximum $P_b(e)$ when $\gamma = q/N$ is varied) behaves like that for partial-band noise jamming, and whether the number of hops per symbol L can be viewed as a form of diversity in mitigating the effects of the jamming.

8.1.4 Jamming Event Probabilities

The general expression (8-19) for the symbol error probability requires the probability $\Pi_L(\underline{\ell})$ of jamming event $\underline{\ell}$, where

$$\underline{\ell} = (\ell_1, \ell_2, \ldots, \ell_M). \tag{8-24}$$

The exact form of $\Pi_L(\underline{\ell})$ will depend upon how the jamming tones are distributed over the band W, i.e. upon the specific jamming models which are discussed in the next section. However, as we show below, if we know the probabilties $\Pi_1(\underline{\ell})$ for L = 1 hop per symbol, then the probabilities $\Pi_L(\underline{\ell})$ for the general case of L hops per symbol may be computed without difficulty by using a recursive technique.

Let $\underline{\ell}_{L_1}$ denote a jamming event taken over a sequence of L_1 hops and $\underline{\ell}_{L_2}$ denote a jamming event taken over a sequence of L_2 hops. Then

$$\underline{\ell}_{L1} = (\ell_1, \ell_2, \ldots, \ell_M) \tag{8-25a}$$

with

$$0 \leq \ell_i' \leq L_1, \quad i = 1, 2 \ldots, M. \tag{8-25b}$$

and

$$\underline{\ell}_{L_2} = (\ell_1', \ell_2', \ldots, \ell_M') \tag{8-26a}$$

with

$$0 \leq \ell_i' \leq L_2, \quad i = 1, 2 \ldots, M. \tag{8-26b}$$

The jamming event $\underline{\ell}_{L_1+L_2}$ taken over the contatenated sequence of $L_1 + L_2$ hops is described by

$$\underline{\ell}_{L_1+L_2} = \underline{\ell}_{L_1} + \underline{\ell}_{L_2}$$

$$= (\ell_1 + \ell_1', \ell_2 + \ell_2', \ldots, \ell_M + \ell_M')$$

$$= (\ell_1'', \ell_2'', \ldots, \ell_M'') \tag{8-27a}$$

where $\ell_i'' \triangleq \ell_i + \ell_i'$, $i = 1, 2, \ldots, M$, and

$$0 \leq \ell_i' \leq L_1 + L_2. \tag{8-27b}$$

Consider a specific instance of the random jamming event $\underline{\ell}_{L_1+L_2}$ say $(\alpha_1, \alpha_1, \ldots, \alpha_M)$ where $0 \leq \alpha_i \leq L_1 + L_2$, $i = 1, 2, \ldots M$. This event may arise in many ways, for the α_i jammed hops in the i-th channel may be apportioned to the segments of length L_1 and L_2 in numerous ways, and each channel i ($i = 1, 2, \ldots, M$) is independently described. More specifically, the number of ways that the α_i jammed hops can be split between the two groups of L_1 and L_2 hops, without regard to the specific order of the jammed hops in each group, is $1 + \min(\alpha_i, L_1, L_2)$. (See Appendix 8A for proof.) The total number of ways that the event $(\alpha_1, \alpha_2, \ldots, \alpha_M)$ may arise, then, is

$$C \triangleq \prod_{i=1}^{M} \left[1 + \min(\alpha_i, L_1, L_2)\right]. \tag{8-28}$$

The probability of the event $(\alpha_1, \alpha_2, \ldots, \alpha_M)$ is the sum of C terms of the form $\Pi_{L_1}(i_1, i_2, \ldots, i_M)\,\Pi_{L_1}(\alpha_1 - i_1, \alpha_2 - i_2, \ldots, \alpha_M - i_M)$ where $0 \leq i_j \leq \alpha_j$, $j = 1, 2, \ldots, M$. This probability may be expressed as

$$\Pi_{L_1+L_2}(\alpha_1, \alpha_1, \ldots, \alpha_M) = \sum_{i_1=\max(0,\, L_2-\alpha_1)}^{\min(\alpha_1,\, L_1)} \cdots \sum_{i_M=\max(0,\, L_2-\alpha_M)}^{\min(\alpha_M,\, L_1)} \Pi_{L_1}(i_1, \ldots, i_M)$$

$$\cdot \Pi_{L_2}(\alpha_1 - i_1, \ldots, \alpha_M - i_M). \tag{8-29}$$

In writing the limits of the summations in (8-29) we have taken into account two conditions:

- if $\alpha_1 > L_2$, then at least $L_2 - \alpha_1$ jammed hops must be apportioned to the L_1 hops, and
- if $\alpha_1 > L_1$, then no more than L_1 jammed hops may be apportioned to the L_1 hops.

The form of each individual summation in (8-29) is that of a discrete convolution operation.

If we let $L_1 = L_2 = 1$ in (8-25)-(8-29), we see that knowledge of Π_1 is sufficient to obtain Π_2. Then with $L_1 = 2$ and $L_2 = 1$ we can obtain Π_3, etc., by repeated applications of (8-29). Therefore, in discussing the particular jamming models in the next section, it will suffice to obtain only $\Pi_1(\underline{\ell})$ for each model. A numerical algorithm for computing the probabilities $\Pi_L(\underline{\ell})$ is outlined in Appendix 8B, and a listing of a computer program implementing this algorithm is given in Appendix 8C.

For the special case of $M = 2$ (binary), the L-fold convolution process may be reduced to a single summation. If the jamming event over L hops is (ℓ_1, ℓ_2) and if exactly $k \leq \min(\ell_1, \ell_2)$ hops are such that both channels are jammed, i.e. event (1,1), then there must be ℓ_1-k hops with per-hop jamming event (1,0) and ℓ_2 - k hops with per-hop jamming event (0,1). Any hops left over must be per-hop jamming event (0,0). Since, furthermore, the number of (0,1) hops plus the number of (1,0) hops cannot exceed the total number of hops, if $\ell_1 + \ell_2 > L$ then k must be at least $\ell_1+\ell_2-L$.

For each value of k, we must consider all possible sequences of hops giving rise to the event (ℓ_1, ℓ_2). This is equivalent to the problem of distributing ℓ_2 indistinguishable red balls and ℓ_2 indistinguishable black balls into L urns such that each urn contains at most one red ball and at most one black ball. The number of ways this can occur is

$$I \triangleq \binom{L}{k,\ \ell_1-k,\ \ell_2-k,\ L-\ell_1-\ell_2+k} \tag{8-30}$$

where k is the number of urns containing both a red ball and a black ball and the multinomial coefficient is defined as

$$\binom{L}{a,\ b,\ c,\ d} = \frac{L!}{a!b!c!d!}\,, \quad a+b+c+d = L. \tag{8-31}$$

Thus we can write the L-hop jamming event probability for the binary case as

$$\Pi_L(\ell_1, \ell_2) = \sum_{k=\max(0,\ell_1+\ell_2-L)}^{\min(\ell_1,\ \ell_2)} \binom{L}{k,\ \ell_1-k,\ \ell_2-k,\ L-\ell_1-\ell_2+k} \left[\Pi_1(1,1)\right]^k$$
$$\cdot \left[\Pi_1(1,0)\right]^{\ell_1-k} \left[\Pi_1(0,1)\right]^{\ell_2-k} \left[\Pi_1(0,0)\right]^{L-\ell_1-\ell_2+k}. \tag{8-32}$$

This form of the expression for $\Pi_L(\ell_1, \ell_2)$ has the advantage of permitting direct evaluation for any L, ℓ_1, and ℓ_2 without need of large intermediate arrays, for the special case of M = 2.

8.2 TONE JAMMING MODELS

The spectrum of the tone jamming waveform consists of a set of discrete spectral lines. Unlike the partial-band noise jammer, a single parameter is inadequate to describe the jammer fully, for we must specify not only a span of frequencies between highest and lowest jammed frequencies (which, alone, was adequate to characterize the noise jammer) but we must also specify how the tones are distributed within the jammed band. This additional freedom in specifying the jamming implies the potential for being many different varieties of tone jamming models.

In the remainder of this section, we describe practical tone jamming models. One common factor is present in all of these models: we assume that the jamming tones coincide exactly in frequency with available signal tone frequencies. This assumption is the same one as was made in Section 8.1.1 to make the analysis more tractable. In addition to describing each of the three models, we also formulate the one-hop jamming event probabilities $\Pi_1(\underline{\ell})$ for each model, since these quantities are required as inputs to the analysis of Section 8.1.4. The models which we consider are

- randomly placed tones,
- evenly spaced tones (barrage jamming), and
- band multitone jamming.

8.2.1 Model 1: Randomly Placed Tones

In this model, the jammer makes q equiprobable random selections, without replacement, from the N hopping slots to determine where to place his q jamming tones, with the constraint $1 \leq q \leq N$. This model has also

been called independent multitone jamming by some authors [13]. Figure 8-3 illustrates this jamming model.

This is the least constrained tone jamming model. On a given hop the number of jamming tones falling within the M-ary symbol may range from 0 to M, and they may be distributed in any pattern within the M cells. However, since the jammer has selected the cells to be jammed with equal probability, the probability of any specific jamming event depends only on the number of cells jammed, and not on the specific arrangement of jammed cells among the M cells of the M-ary symbol.

Let the jamming event on a specific hop be denoted by $\underline{\nu} \triangleq (\nu_1, \nu_2, \ldots, \nu_M)$ where

$$\nu_k = \begin{cases} 0, & \text{kth filter unjammed} \\ 1, & \text{kth filter jammed,} \quad k = 1, 2, \ldots, M. \end{cases} \tag{8-33}$$

The total number of filters containing jamming tones is

$$\ell = \sum_{k=1}^{M} \nu_k . \tag{8-34}$$

Obviously, $0 \leq \ell \leq M$. The probability $\Pi_1(\underline{\nu})$ of the occurrence of jamming event $\underline{\nu}$ is

$$\Pi_1(\underline{\nu}) = \frac{q}{N} \cdot \frac{q-1}{N-1} \cdot \ldots \cdot \frac{q-\ell+1}{N-\ell+1} \cdot \frac{N-q+\ell-1}{N-\ell} \cdot \frac{N-q+\ell-2}{N-\ell-1} \cdot \ldots \cdot \frac{N-q-M+1}{N-M+1} \tag{8-35a}$$

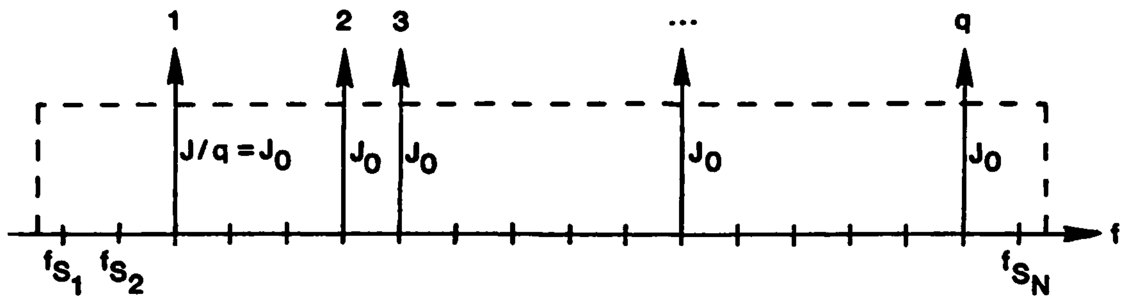


FIGURE 8-3 RANDOMLY PLACED JAMMING TONES (INDEPENDENT MULTITONE JAMMING)

or, more compactly,

$$\Pi_1(\underline{\nu}) = \prod_{k=1}^{\ell} \frac{q-k+1}{N-k+1} \prod_{k=\ell+1}^{M} \frac{N-q-k+1}{N-k+1} \tag{8-35b}$$

which can also be written in the form

$$\Pi_1(\underline{\nu}) = \frac{(q-\ell+1)_\ell \, (N-q-M+1)_{M-\ell}}{(N-M+1)_M} \tag{8-35c}$$

where we have used the Pochhammer notation

$$(a)_n = \frac{\Gamma(a+n)}{\Gamma(a)}, \ a \neq 0 \tag{8-35d}$$

$$(0)_n = \begin{cases} 0, & n \neq 0 \\ 1, & n = 0. \end{cases} \tag{8-35e}$$

Several other equivalent formulations of the expression for $\Pi_1(\underline{\nu})$ may be written using relations between the Pochhammer notation and the binomial coefficients. A selection of these alternative formulations is presented in Appendix 8D.

8.2.2 Model 2: Barrage Jamming

Rather than randomly selecting the location for each jamming tone, the jammer may choose to employ a more structured approach to distributing the q available jamming tones over the N frequency hopping cells used by the communicator. One such more structured approach is that which we have called barrage jamming. The barrage jammer makes a random choice of the location for the first jamming tones. The jammer then spaces the remaining q-1 tones at intervals of n slots above or below the first tone, as shown in Figure 8-4. This method of selecting the locations of the q

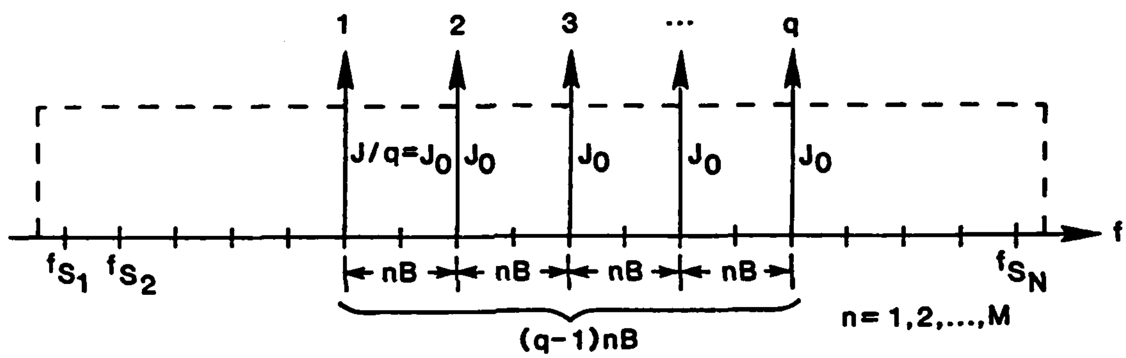


FIGURE 8-4 BARRAGE JAMMING (EVENLY SPACED TONES)

available jamming tones gives the barrage jammer more control over how many of the jamming tones will fall within a jammed M-ary symbol.

Let the total number of hopping slots available to the communicator be N, each of which has a bandwidth of B Hz. The total spread spectrum bandwidth then is W=NB Hz, and the fraction of the band jammed is $\gamma = q/N$. Since the jamming tones are spaced at intervals of n slots, the total size of the jammed portion of the bandwidth is (q-1)nB+B Hz. The size must not be greater than W if all jamming tones are to be effective against the targeted communications system. Therefore, the number of tones which the jammer may effectively employ is upper bounded by

$$q \leq \frac{N-1}{n} + 1. \tag{8-36}$$

Assume that K of the available jamming tones hit the M-ary symbol cluster on a given hop. The possible values of K are limited to

$$K \leq \min\left(q, \left\lceil \frac{M}{n} + 1 \right\rceil\right) \tag{8-37}$$

where the notation $\lceil x \rceil$ denotes the smallest integer which is greater than or equal to x. From (8-37), we observe that when $n \geq M$, the maximum number of tones hitting the M-ary symbol cluster is one. This is the case where the jammer wants to hit the M-ary symbol cluster with either exactly one tone or none at all. Therefore, we need to consider only those cases for which $1 \leq n \leq M$.

When no constraint is imposed on the distribution of the jamming tones (as was the case for randomly located tones), the maximum number of possible jamming events for a single hop is 2^M. However, when the constraint of evenly spaced jamming tones is imposed on the barrage jammer, the number of possible events is 2M-(n-1). For large M, this is a significant reduction

in the number of possible jamming events under barrage jamming, as compared to randomly located jamming tones: the growth of the number of distinguishable jamming events is linear in M, rather than exponential in M. For example, when M = 16, the distinguishable jamming events for randomly located tones is 2^{16} = 65,536, whereas for barrage jamming with n=1 for the spacing parameter, there are only 2×16-(1-1) = 32 jamming events possible.

Assuming that the jamming tones do not occupy any of the hopping slots within M slots of the upper and lower edges of the system bandwidth W (i.e., there is room for a symbol to lie outside the jammed fraction of the band on both sides of the portion of the band spanned by the jamming tones), we determine the number H(K) of possible hopping positions at which K of the M cells available to the communicator are jammed by the barrage jammer with spacing parameter n. The algorithm for computing H(K) is given by the flow diagram of Figure 8-5.

For jamming tones spaced at intervals of n hopping cells, there are 2M-(n-1) = 2M-n+1 distinguishable jamming events under barrage jamming. Using the algorithm given in Figure 8-5, we can compute the jamming event probabilities $\Pi_1(\underline{\nu})$ for various combinations of the parameters M and n as a function of the quantities N and q. The results of numerical calculations of $\Pi_1(\underline{\nu})$ are given in Table 8-1 which lists all possible jamming events and their associated probabilities for M=2, 4, and 8 and $1 \leq n \leq M$. In this table, the jamming-event vector $\underline{\nu}$ = (a,b,c,...), where the elements a, b, c, ... take on the values 0 or 1. If an event is not listed in the table, then the associated probability is zero.

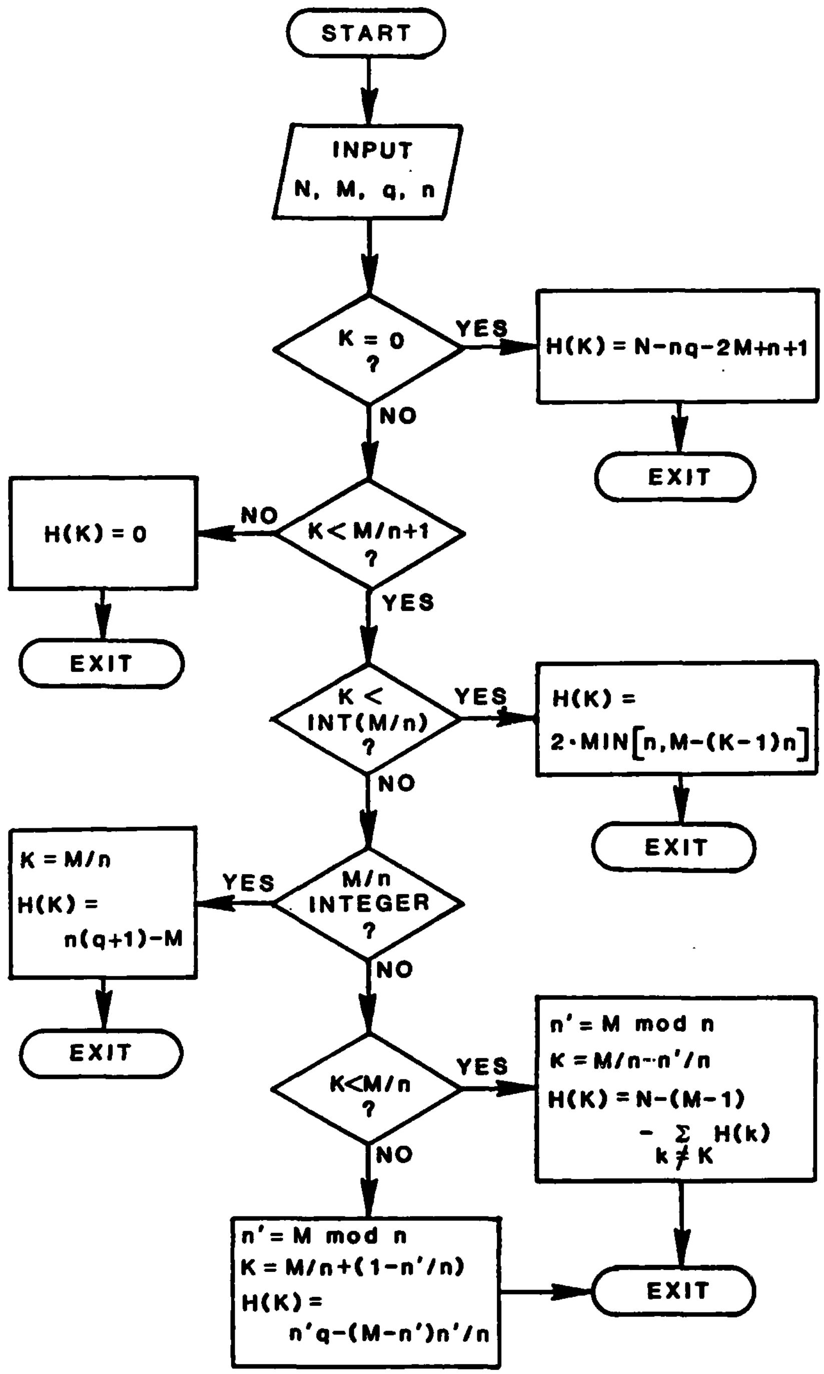


FIGURE 8-5 ALGORITHM TO COMPUTE THE NUMBER OF POSSIBLE POSITIONS H(K) FOR K JAMMING TONES IN THE M-ARY BAND WHEN q JAMMING TONES ARE SPACED AT INTERVALS OF nB Hz

TABLE 8-1
BARRAGE JAMMING EVENTS WITH NON-ZERO PROBABILITY FOR L=1 HOP/SYMBOL

(a) M=2, n=1

a,b	Pr(a,b)
0,0	(N-q-2)/(N-1)
0,1	1/(N-1)
1,0	1/(N-1)
1,1	(q-1)/(N-1)

(b) M=2, n=2

a,b	Pr(a,b)
0,0	(N-2q-1)/(N-1)
0,1	q/(N-1)
1,0	q/(N-1)

(c) M=4, n=1

a,b,c,d	Pr(a,b,c,d)
0,0,0,0	(N-q-6)/(N-3)
0,0,0,1	1/(N-3)
0,0,1,1	1/(N-3)
0,1,1,1	1/(N-3)
1,0,0,1	1/(N-3)
1,0,0,0	1/(N-3)
1,1,0,0	1/(N-3)
1,1,1,0	1/(N-3)
1,1,1,1	(q-3)/(N-3)

(d) M=4, n=2

a,b,c,d	Pr(a,b,c,d)
0,0,0,0	(N-2q-5)/(N-3)
0,0,0,1	1/(N-3)
0,0,1,0	1/(N-3)
0,1,0,0	1/(N-3)
0,1,0,1	(q-1)/(N-3)
1,0,0,0	1/(N-3)
1,0,1,0	(q-1)/(N-3)

(e) M=4, n=3

a,b,c,d	Pr(a,b,c,d)
0,0,0,0	(N-3q-4)/(N-3)
0,0,0,1	1/(N-3)
0,0,1,0	q/(N-3)
0,1,0,0	q/(N-3)
1,0,0,0	1/(N-3)
1,0,0,1	(q-1)/(N-3)

(f) M=4, n=4

a,b,c,d	Pr(a,b,c,d)
0,0,0,0	(N-4q-3)/(N-3)
0,0,0,1	q/(N-3)
0,0,1,0	q/(N-3)
0,1,0,0	q/(N-3)
1,0,0,0	q/(N-3)

TABLE 8-1 (Cont.)

(g) M=8, n=1

a,b,c,d,e,f,g,h	Pr(a,b,c,d,e,f,g,h)
0,0,0,0,0,0,0,0	(N-q-14)/(N-7)
0,0,0,0,0,0,0,1	1/(N-7)
0,0,0,0,0,0,1,1	1/(N-7)
0,0,0,0,0,1,1,1	1/(N-7)
0,0,0,0,1,1,1,1	1/(N-7)
0,0,0,1,1,1,1,1	1/(N-7)
0,0,1,1,1,1,1,1	1/(N-7)
0,1,1,1,1,1,1,1	1/(N-7)
1,0,0,0,0,0,0,0	1/(N-7)
1,1,0,0,0,0,0,0	1/(N-7)
1,1,1,0,0,0,0,0	1/(N-7)
1,1,1,1,0.0,0,0	1/(N-7)
1,1,1,1,1,0,0,0	1/(N-7)
1,1,1,1,1,1,0,0	1/(N-7)
1,1,1,1,1,1,1,0	1/(N-7)
1,1,1,1,1,1,1,1	(q-7)/(N-7)

(h) M=8, n=2

a,b,c,d,e,f,g,h	Pr(a,b,c,d,e,f,g,h)
0,0,0,0,0,0,0,0	(N-2q-13)/(N-7)
0,0,0,0,0,0,0,1	1/(N-7)
0,0,0,0,0,0,1,0	1/(N-7)
0,0,0,0,0,1,0,1	1/(N-7)
0,0,0,0,1,0,1,0	1/(N-7)
0,0,0,1,0,1,0,1	1/(N-7)
0,0,1,0,1,0,1,0	1/(N-7)
0,1,0,0,0,0,0,0	1/(N-7)
0,1,0,1,0,0,0,0	1/(N-7)
0,1,0,1,0,1,0,0	1/(N-7)
0,1,0,1,0,1,0,1	(q-3)/(N-7)
1,0,0,0,0,0,0,0	1/(N-7)
1,0,1,0,0,0,0,0	1/(N-7)
1,0,1,0,1,0,0,0	1/(N-7)
1,0,1,0,1,0,1,0	(q-3)/(N-7)

(i) M=8, n=3

a,b,c,d,e,f,g,h	Pr(a,b,c,d,e,f,g,h)
0,0,0,0,0,0,0,0	(N-3q-12)/(N-7)
0,0,0,0,0,0,0,1	1/(N-7)
0,0,0,0,0,0,1,0	1/(N-7)
0,0,0,0,0,1,0,0	1/(N-7)
0,0,0,0,1,0,0,1	1/(N-7)
0,0,0,1,0,0,1,0	1/(N-7)
0,0,1,0,0,0,0,0	1/(N-7)
0,0,1,0,0,1,0,0	(q-1)/(N-7)
0,1,0,0,0,0,0,0	1/(N-7)
0,1,0,0,1,0,0,0	1/(N-7)
0,1,0,0,1,0,0,1	(q-2)/(N-7)
1,0,0,0,0,0,0,0	1/(N-7)
1,0,0,1,0,0,0,0	1/(N-7)
1,0,0,1,0,0,1,0	(q-2)/(N-7)

(j) M=8, n=4

a,b,c,d,e,f,g,h	Pr(a,b,c,d,e,f,g,h)
0,0,0,0,0,0,0,0	(N-4q-11)/(N-7)
0,0,0,0,0,0,0,1	1/(N-7)
0,0,0,0,0,0,1,0	1/(N-7)
0,0,0,0,0,1,0,0	1/(N-7)
0,0,0,0,1,0,0,0	1/(N-7)
0,0,0,1,0,0,0,0	1/(N-7)
0,0,0,1,0,0,0,1	(q-1)/(N-7)
0,0,1,0,0,0,0,0	1/(N-7)
0,0,1,0,0,0,1,0	(q-1)/(N-7)
0,1,0,0,0,0,0,0	1/(N-7)
0,1,0,0,0,1,0,0	(q-1)/(N-7)
1,0,0,0,0,0,0,0	1/(N-7)
1,0,0,0,1,0,0,0	(q-1)/(N-7)

TABLE 8-1 (Concluded)

(k) M=8, n=5

a,b,c,d,e,f,g,h	Pr(a,b,c,d,e,f,g,h)
0,0,0,0,0,0,0,0	(N-5q-10)/(N-7)
0,0,0,0,0,0,0,1	1/(N-7)
0,0,0,0,0,0,1,0	1/(N-7)
0,0,0,0,0,1,0,0	1/(N-7)
0,0,0,0,1,0,0,0	q/(N-7)
0,0,0,1,0,0,0,0	q/(N-7)
0,0,1,0,0,0,0,0	1/(N-7)
0,0,1,0,0,0,0,1	(q-1)/(N-7)
0,1,0,0,0,0,0,0	1/(N-7)
0,1,0,0,0,0,1,0	(q-1)/(N-7)
1,0,0,0,0,0,0,0	1/(N-7)
1,0,0,0,0,1,0,0	(q-1)/(N-7)

(l) M=8, n=6

a,b,c,d,e,f,g,h	Pr(a,b,c,d,e,f,g,h)
0,0,0,0,0,0,0,0	(N-6q-9)/(N-7)
0,0,0,0,0,0,0,1	1/(N-7)
0,0,0,0,0,0,1,0	1/(N-7)
0,0,0,0,0,1,0,0	q/(N-7)
0,0,0,0,1,0,0,0	q/(N-7)
0,0,0,1,0,0,0,0	q/(N-7)
0,0,1,0,0,0,0,0	q/(N-7)
0,1,0,0,0,0,0,0	1/(N-7)
0,1,0,0,0,0,0,1	(q-1)/(N-7)
1,0,0,0,0,0,0,0	1/(N-7)
1,0,0,0,0,0,1,0	(q-1)/(N-7)

(m) M=8, n=7

a,b,c,d,e,f,g,h	Pr(a,b,c,d,e,f,g,h)
0,0,0,0,0,0,0,0	(N-7q-8)/(N-7)
0,0,0,0,0,0,0,1	1/(N-7)
0,0,0,0,0,0,1,0	q/(N-7)
0,0,0,0,0,1,0,0	q/(N-7)
0,0,0,0,1,0,0,0	q/(N-7)
0,0,0,1,0,0,0,0	q/(N-7)
0,0,1,0,0,0,0,0	q/(N-7)
0,1,0,0,0,0,0,0	q/(N-7)
1,0,0,0,0,0,0,0	1/(N-7)
1,0,0,0,0,0,0,1	(q-1)/(N-7)

(n) M=8, n=8

a,b,c,d,e,f,g,h	Pr(a,b,c,d,e,f,g,h)
0,0,0,0,0,0,0,0	(N-8q-7)/(N-7)
0,0,0,0,0,0,0,1	q/(N-7)
0,0,0,0,0,0,1,0	q/(N-7)
0,0,0,0,0,1,0,0	q/(N-7)
0,0,0,0,1,0,0,0	q/(N-7)
0,0,0,1,0,0,0,0	q/(N-7)
0,0,1,0,0,0,0,0	q/(N-7)
0,1,0,0,0,0,0,0	q/(N-7)
1,0,0,0,0,0,0,0	q/(N-7)

8.2.3 Model 3: Band Multitone Jamming

The band multitone jammer, as defined by Levitt [13], distributes the q available jamming tones over the communicator's spread bandwidth W in such a way that the M-ary symbol bandwidth contains exactly n jamming tones*, or none. This is illustrated in Figure 8-6. Implementation of this strategy, in general, assumes that the frequency hopping system being jammed employs distinct non-overlapping M-ary bands and that the locations of these bands are known to the jammer [13, 26, 27]. In this respect, we must modify our earlier model of the spread spectrum system. For the communicator's possible M-ary bands to be non-overlapping, there can be only N/M possible hopping positions, and we are constrained to have N an integer multiple of M, or equivalently to have W an integer multiple of MB.

The number of M-ary bands jammed by the band multitone jammer is q/n, where each band contains exactly n jamming tones. Therefore the probability that the M-ary band contains n jamming tones (without regard to the arrangement of jammed and unjammed signalling frequencies within the M-ary band) on a given hop is

$$\Pr\{\text{jammed}\} = \frac{\text{number of jammed bands}}{\text{total number of bands}} = \frac{q/n}{N/M} = \frac{qM}{Nn} \tag{8-38}$$

and the probability that the M-ary band is unjammed is

$$\Pr\{\text{unjammed}\} = 1 - \Pr\{\text{jammed}\} = 1 - \frac{qM}{Nn}\,. \tag{8-39}$$

* We emphasize that the parameter n defined here is quite different from the parameter n for the barrage jammer.

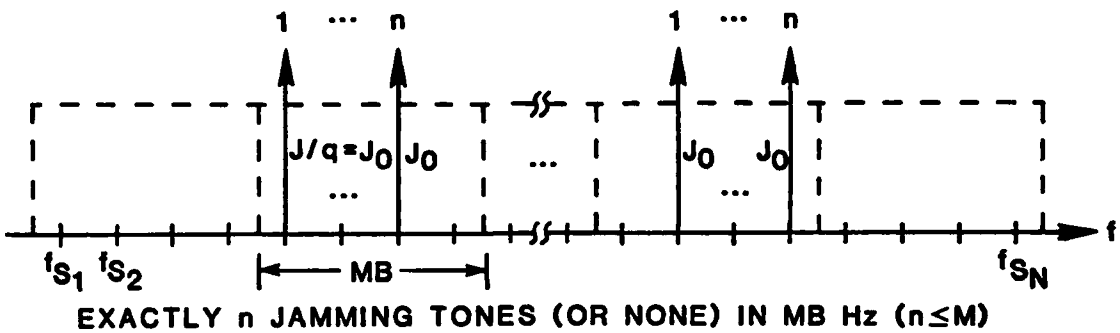


FIGURE 8-6 BAND MULTITONE JAMMING

The n jamming tones in an M-ary band can be distributed over the band $\binom{M}{n}$ ways. If we assume that each of these arrangements is equally probable, then we have for band multitone jamming

$$\Pi_1(\underline{\nu}) = \begin{cases} 1 - \frac{qM}{Nn}, & \sum_{i=1}^{M} \nu_i = 0 \\ \frac{qM}{Nn\binom{M}{n}}, & \sum_{i=1}^{M} \nu_i = n \\ 0, & \text{otherwise} \end{cases} \tag{8-40}$$

where $\underline{\nu} = (\nu_1, \nu_2, \ldots, \nu_M)$, $\nu_i \varepsilon \{0, 1\}$, $i = 1, \ldots, M$, is a 1-hop jamming event as defined in (8-1).

As an exception to the above discussion, we observe that band multitone jamming with n = 1 tone per symbol band is equivalent to barrage jamming with n = M cells spacing between tones. In this particular case, overlapping hopping positions may be allowed when considering band multitone jamming.

8.3 NUMERICAL RESULTS

We now turn our consideration to numerical evaluation of the bit error probabilities for various combinations of M, L, and jamming models. The equation for $P_b(e)$ as a function of the various parameters is quite complicated and involves two numerical integrations. Therefore, our first concern is how to compute the probabilities in an efficient manner. After discussing the computational procedures, we present and discuss the numerical results obtained by these procedures.

8.3.1 Computational Procedures

As we see from (8-19), the computation of the total error probabiliity requires us to average the conditional error probabilities over all jamming events. For a system employing L hops/symbol with an alphabet of M symbols, a jamming event is described by the M-tuple $(\ell_1, \ell_2, \ldots, \ell_M)$ where $0 \leq \ell_i \leq L$, $i = 1, 2, \ldots, M$. The total number of possible jamming events, then, is $(L+1)^M$. As shown in Table 8-2, this number rapidly becomes quite large as L and M increase. Of course, not all jamming events can occur for the band multitone and barrage jamming models; but a computer program would, at least, have to consider each event and determine if it could occur. If the computer could test 100,000 events per second against the criteria for having a non-zero $\Pi_L(\underline{\ell})$, than for M = 16 and L = 6 it would take 3.32×10^8 seconds $\approx$ 10.5 years just to examine the list of jamming events. Clearly, we cannot use such techniques for large L and/or M; therefore, we restrict our attention to $M \leq 8$ and $L \leq 3$. Even then, computer time considerations have led us to omit the case M=8, L=3 from numerical examination.

In a second step towards reducing the computational load, we have sought other expressions, either special cases or approximations, which may be computed more efficiently. We have successfully used both special case equations and approximations in obtaining numerical results.

8.3.1.1 Special Case Equations for L = 1 Hop/Symbol When There Is Only One Jamming Tone in the Symbol Band

If we restrict our attention to one hop per symbol, the analysis is greatly simplified. If we further confine our considerations to those

TABLE 8-2

NUMBER OF JAMMING EVENTS AS A FUNCTION OF ALPHABET SIZE AND NUMBER OF HOPS PER SYMBOL

L HOPS/SYMBOL	ALPHABET SIZE, M			
	2	4	8	16
1	4	16	256	65,536
2	9	81	6,561	43,046,721
3	16	256	65,536	4,294,967,296
4	25	625	390,625	152,587,890,625
5	36	1,296	1,679,616	2,821,109,907,456
6	49	2,401	5,764,801	33,232,930,569,601

jamming models which permit at most one jamming tone per M-ary symbol (i.e. barrage jamming for $n = M$ cells spacing, band multitone jamming for $n = 1$ tone per cell, or any model for the special case of $q = 1$ tone in the system bandwidth W), a closed-form solution is available for the conditional error probabilities [14]. When the jamming tone and the signal tone fall in the same filter,

P_s(e|jamming and signal in the same filter) =

$$\sum_{m=1}^{M-1} (-1)^{m+1} \binom{M+1}{m} \frac{1}{m+1} I_0\left[\frac{2\sqrt{SJ_0}}{\sigma_N^2} \left(\frac{m}{m+1}\right)\right] \exp\left[\frac{(S+J_0)\, m}{\sigma_N^2\,(m+1)}\right] \tag{8-41}$$

and when the jamming tone and the signal fall into different filters,

P_s(e|jamming and signal in different filters) =

$$1 + \sum_{m=0}^{M-2} (-1)^{m+1} \binom{M-2}{m} \frac{1}{m+1} \exp[-m(m+2)b]$$

$$\cdot\left\{1 - Q(\sqrt{2a}, \sqrt{2b}\,) + \frac{1}{m+2} \exp[-(a+b)]\; I_0(2\sqrt{ab}\,)\right\} \tag{8-42}$$

where

$$a \triangleq \frac{J_0}{\sigma_N^2} \left(\frac{m+1}{m+2}\right) \tag{8-43}$$

and

$$b \triangleq \frac{S}{\sigma_N^2} \frac{1}{(m+1)(m+2)}. \tag{8-44}$$

For the case of no jamming tones in the M-ary band, the conditional error probability is the conventional M-ary error probability [17, eq. 14-45]

$$P_s(e|0, 0,\ldots, 0) = \sum_{m=1}^{M-1} \binom{M+1}{m} \frac{(-1)^{m+1}}{m+1} \exp\left[-\frac{mS}{(m+1)\sigma_N^2}\right]. \quad (8\text{-}45)$$

8.3.1.2 Approximation to Error-Rate Expression for General Jamming Model

Although the forms obtained in (8-41)-(8-45) are easy to compute, they cannot be applied to the general jamming models for all L. Therefore, we have examined several approximations to the general form of the error probability equation. As shown in Appendix 8E, we can approximate the density of the signal-channel output by

$$p_{z_1}(\alpha|\ell_1) \approx \frac{1}{2} \exp\left(-\frac{\alpha+a}{2}\right)\left(\frac{\alpha}{a}\right)^{(L-1)/2} I_{L-1}(\sqrt{a\alpha}) \quad (8\text{-}46)$$

where

$$a = 2\left(\frac{E_b}{N_0}\right)\left[K + \frac{\ell_1}{\gamma(E_b/N_J)}\right]. \quad (8\text{-}47)$$

Using this approximate form in (8-8) yields the form

$$P_s(e|\underline{\ell}) \approx \int_0^\infty dy\; p_{z_1}(\sigma_N^2 y|\ell_1) \left\{1 - \prod_{m=2}^{M} [1 - Q_L(\sqrt{\lambda_m}, \sqrt{y})]\right\} \quad (8\text{-}48)$$

in which only one integral remains to be done numerically. The elimination of a numerical integral over the density of the signal-jammer phase differences greatly simplifies the computational task, especially when one considers that the phase-difference density itself must be computed by numerical integration for $L \geq 3$ hops/symbol.

To assess the accuracy of this approximation, we show in Figure 8-7 a comparison of results obtained for M = 2, L = 1 hop/symbol, N = 2400 slots, and E_b/N_0 = 13.3525 dB under barrage jamming with spacing n = 2 slots. In the figure the solid lines represent results computed using the result (8-18) of the exact analysis and the dashed curves represent results obtained using (8-46)-(8-48). We see that the agreement between the result from the approximate form and the exact result is quite good for this case. This gives us confidence that comparisons of jamming models using the approximate form for efficiency will be valid.

8.3.2 Comparison of Jamming Models

Because of the large number of cases to be considered, we used the approximate formulas (8-46)-(8-48) to compare the effectiveness of the several jamming models under consideration. To keep the size of the computational task within reasonable bounds, we also have restricted the numerical comparisons to the case M = 2 and L = 1, 2, and 4. The computer program for these calculations is given in Appendix 8F.

On the basis of these comparisons, we select the most effective jamming models for further study using the exact formulation for the bit error probability for higher values of M.

8.3.2.1 Effects of Randomly Placed Jamming Tones

Figure 8-8 through 8-10 show the bit error probability as a function of the bit energy-to-jamming density ratio for BFSK/FH with L = 1, 2, and 4 hops/bit, respectively, and the partial-band jamming fraction $\gamma = q/N$ as a parameter. In the figures we have also shown in a dashed line the envelope of the curves, which represents the optimum choice of γ from the jammer's viewpoint.

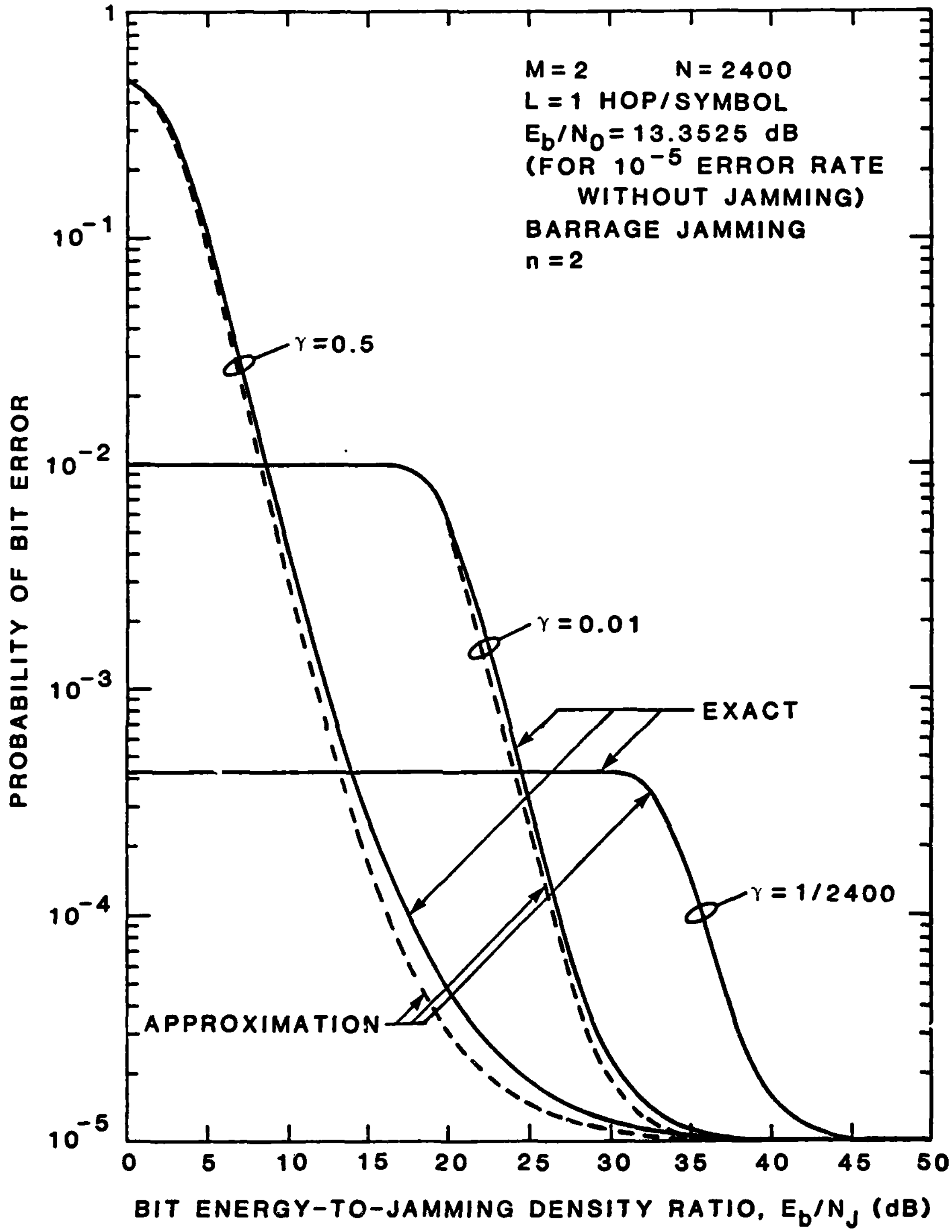


FIGURE 8-7 EXAMPLE COMPARISON OF APPROXIMATE AND EXACT RESULTS FOR TONE JAMMING

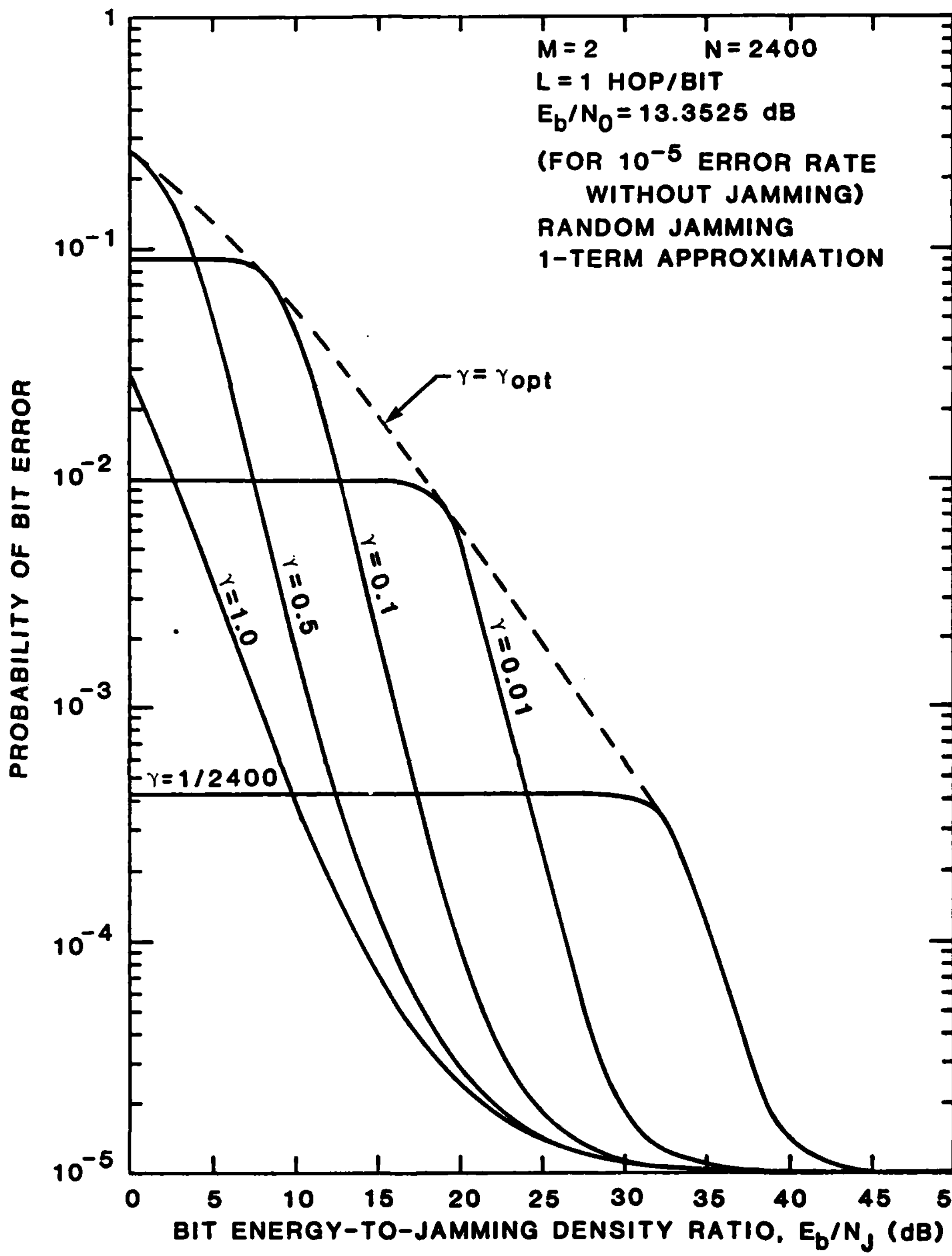


FIGURE 8-8 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR INDEPENDENT MULTITONE JAMMING WHEN L = 1 HOP/BIT AND N = 2400 HOPPING SLOTS FOR BFSK/FH WITH E_b/N_0 = 13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

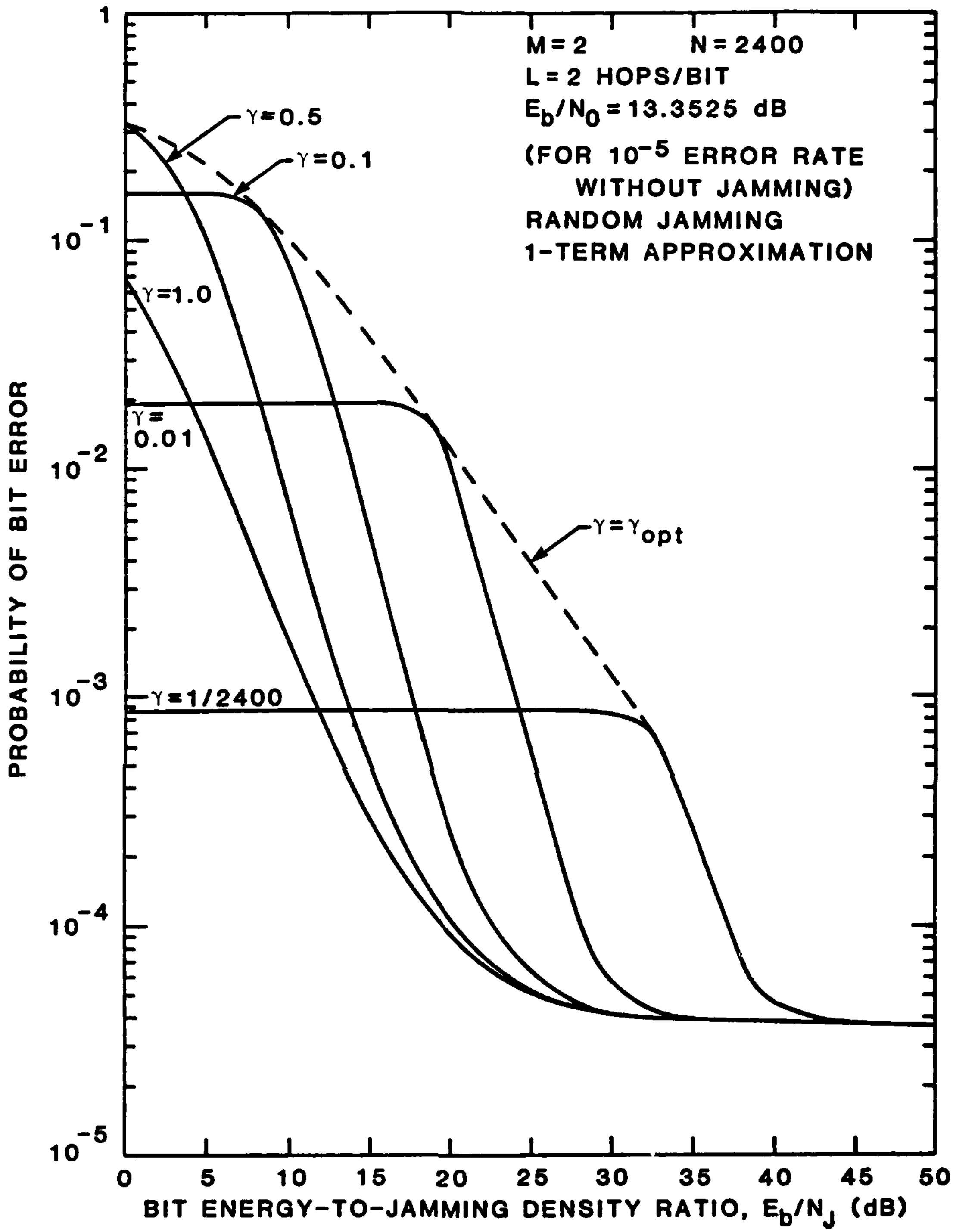


FIGURE 8-9 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR INDEPENDENT MULTITONE JAMMING WHEN L = 2 HOPS/BIT AND N = 2400 HOPPING SLOTS FOR BFSK/FH WITH E_b/N_0 = 13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

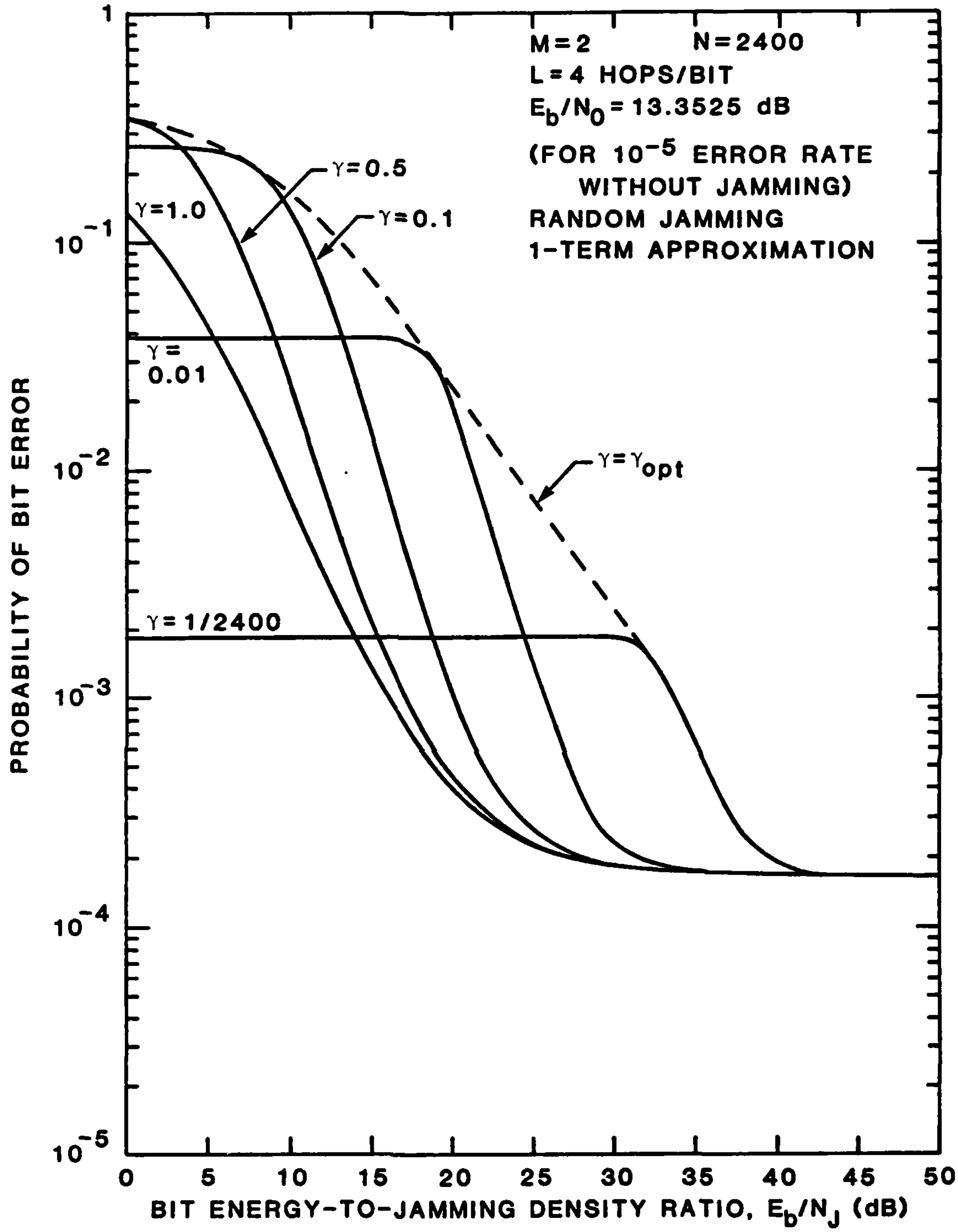


FIGURE 8-10 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR INDEPENDENT MULTITONE JAMMING WHEN L=4 HOPS/BIT AND N=2400 HOPPING SLOTS FOR BFSK/FH WITH E_b/N_0=13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

From Figure 8-8 we note that all the curves, regardless of γ, approach 10^{-5} BER asymptotically for high values of E_b/N_J, as is expected when the jamming becomes negligible. At the other extreme, the jamming becomes very strong and the curves approach another asymptote, which is predicted by the form of the error rate expressions, as follows. If $J \gg S$, then the conditional error probabilites are approximately

$$P_b(e|0, 0) = \text{unjammed error probability} = 10^{-5} \tag{8-49a}$$

$$P_b(e|0, 1) \simeq 1 \tag{8-49b}$$

$$P_b(e|1, 0) \simeq 0 \tag{8-49c}$$

$$P_b(e|1, 1) \simeq 1/2 \tag{8-49d}$$

since the signal is negligible compared to the jamming. Then from (8-19) we have

$$P_b(e) \simeq 10^{-5}\,\Pi_1(0, 0) + \Pi_1(0, 1) + \frac{1}{2}\,\Pi_1(1, 1). \tag{8-50}$$

The first term of (8-50) is very small compared to the other terms as long as $\Pi_1(0, 1) \gg 10^{-5}$ and $\Pi_1(1, 1) \gg 10^{-5}$, and thus we may write

$$P_b(e) \simeq \Pi_1(0, 1) + \frac{1}{2}\,\Pi_1(1, 1). \tag{8-51}$$

Similar asymptotic behavior is seen in Figures 8-9 and 8-10 for L = 2 and L = 4 hops/bit, respectively. In these figures the asymptote in the thermal-noise-limited region (high E_b/N_J) is greater than 10^{-5}, due to noncoherent combining loss, as was discussed in Section 2.1.2. In the jamming-limited region (low E_b/N_J), the asymptote may be obtained in a manner similar to (8-49) and (8-50) using the $\Pi_L(\underline{\ell})$ as appropriate and assuming that

$$P_b(e|\ell_1, \ell_2) \approx \begin{cases} 0, & \ell_1 > \ell_2 \\ \tfrac{1}{2}, & \ell_1 = \ell_2 \\ 1, & \ell_1 < \ell_2. \end{cases} \tag{8-52}$$

The effect of independent multitone jamming, i.e. randomly placed jamming tones, is summarized in Figure 8-11 which shows the optimum jamming fraction envelopes from Figures 8-8 through 8-10 on a common plot. We observe that increasing L consistently degrades the communications link. This is due to a combination of two factors. First, the noncoherent combining loss degrades the link, even in the absence of jamming. Second, the multiple hops give additional opportunities for the signal to hop into the jammed part of the band and suffer degradation from a strong jamming tone.

In Figures 8-8 through 8-11, we note that the optimum-γ curve merges with the $\gamma = 1/N$ curve, since physically there can be only an integer number of jamming tones. Once the optimum value of γ reaches 1/N, it can decrease no further.

8.3.2.2 Effects of Barrage Jamming

For the case of BFSK/FH, i.e. M = 2, we have two variations on the barrage jamming model, as pointed out in Section 8.2.2, for the tone-spacing parameter n = 1 and n = 2. Each of these cases is discussed separately.

For spacing n = 1, i.e. the q tones are located in a group of q contiguous slots, the bit error performance of the communicator's link is shown in Figures 8-12 through 8-14 for L = 1, L = 2, and L = 4 hops/bit, respectively. Again, we have taken the jamming fraction γ as a parameter. We observe from these curves that the optimum-γ envelope does not exhibit

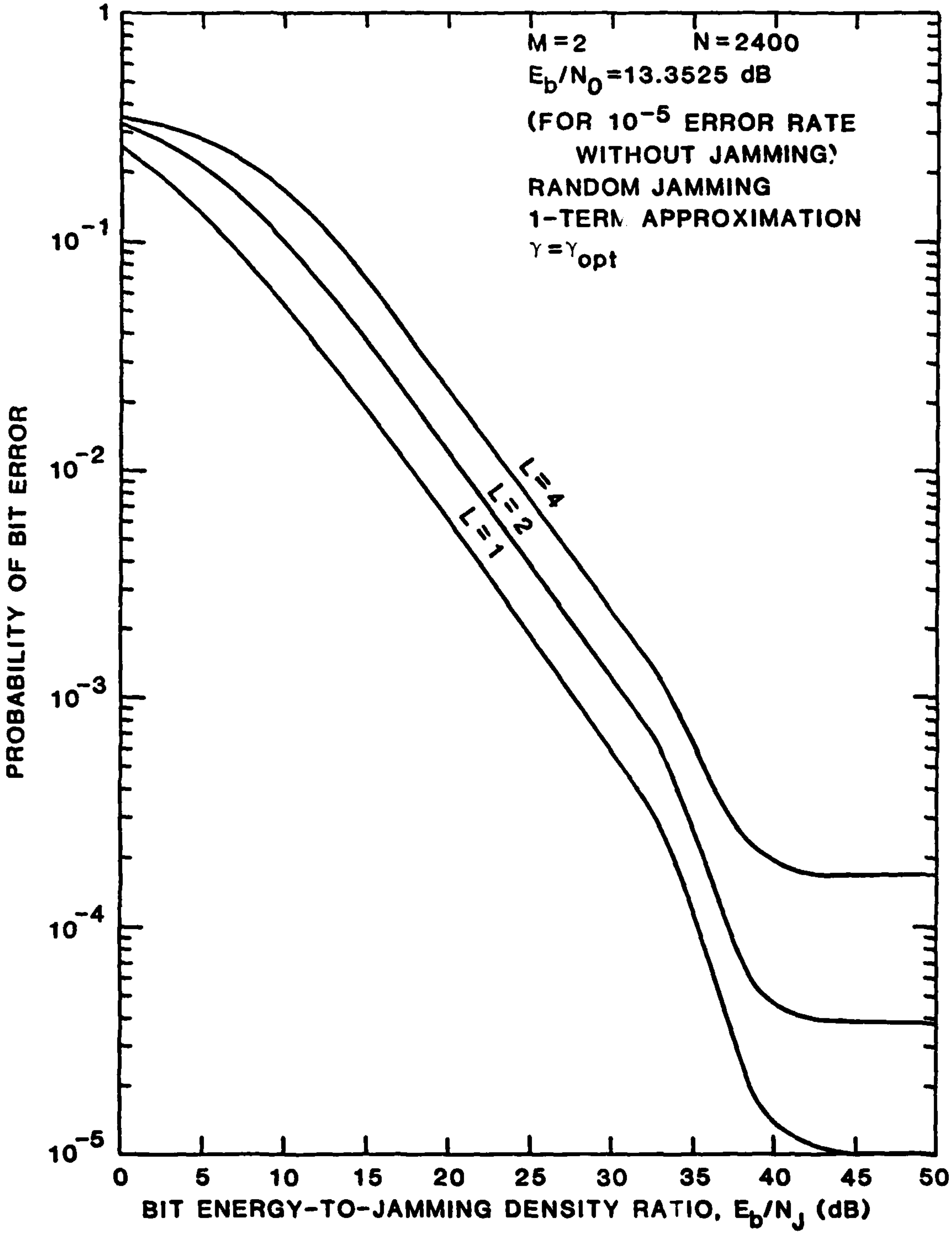


FIGURE 8-11 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH NUMBER OF HOPS PER BIT AS A PARAMETER FOR INDEPENDENT MULTITONE JAMMING WITH N=2400 HOPPING SLOTS AND OPTIMUM JAMMING FRACTION FOR BFSK/FH WITH E_b/N_0=13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

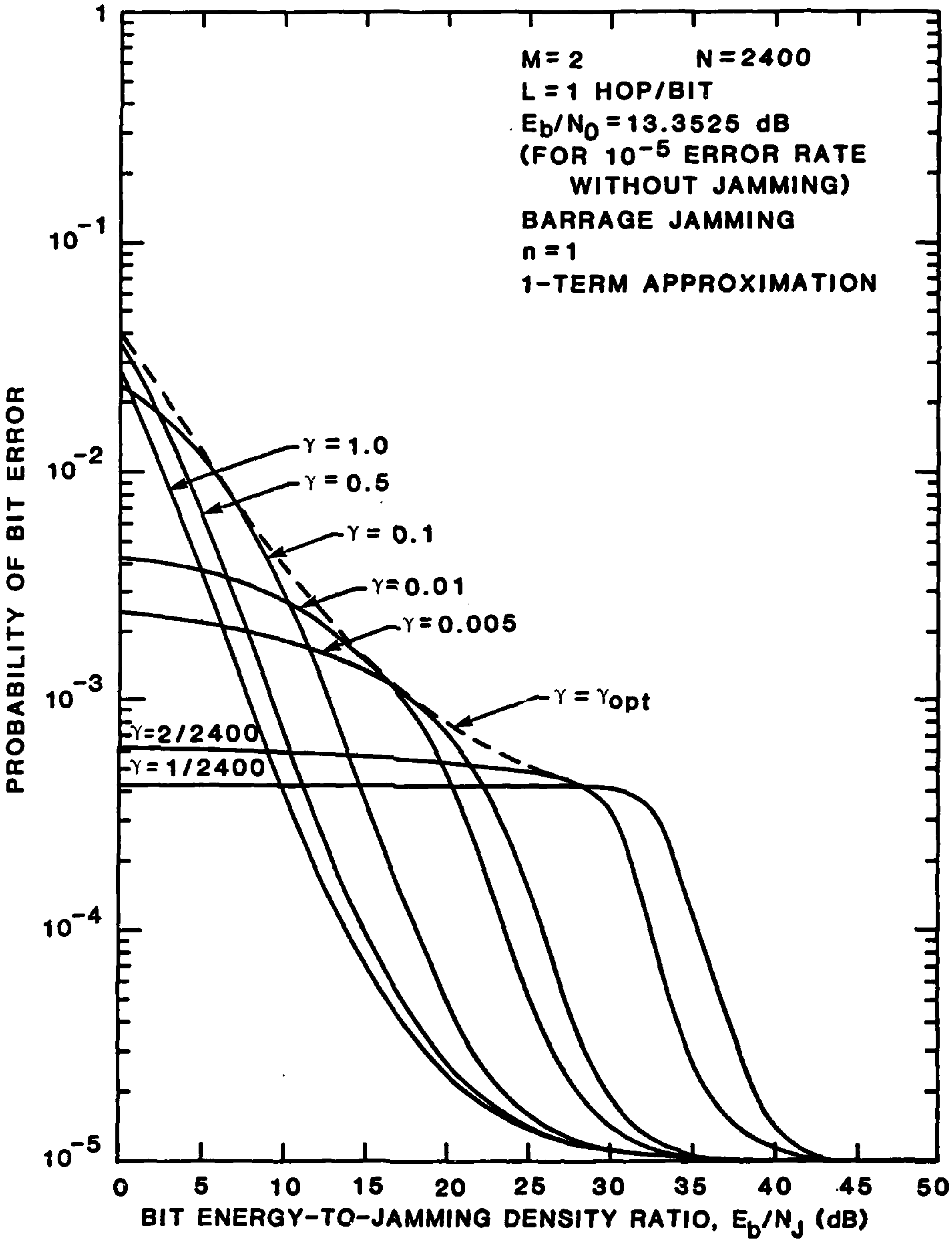


FIGURE 8-12 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n = 1) JAMMING WHEN L = 1 HOP/BIT AND N = 2400 HOPPING SLOTS FOR BFSK/FH WITH E_b/N_0 = 13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

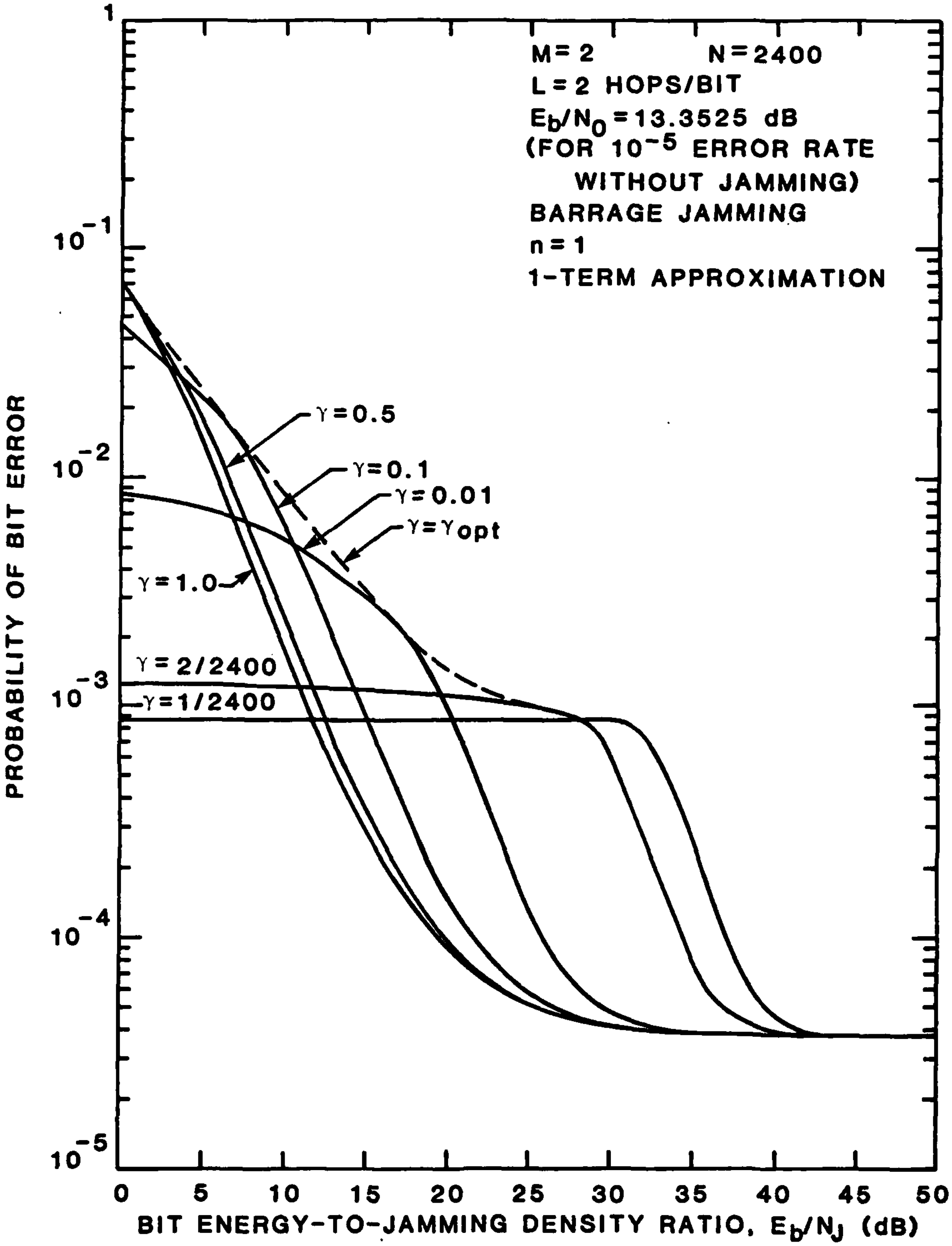


FIGURE 8-13 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n = 1) JAMMING WHEN L = 2 HOPS/BIT AND N = 2400 HOPPING SLOTS FOR BFSK/FH WITH E_b/N_0 = 13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

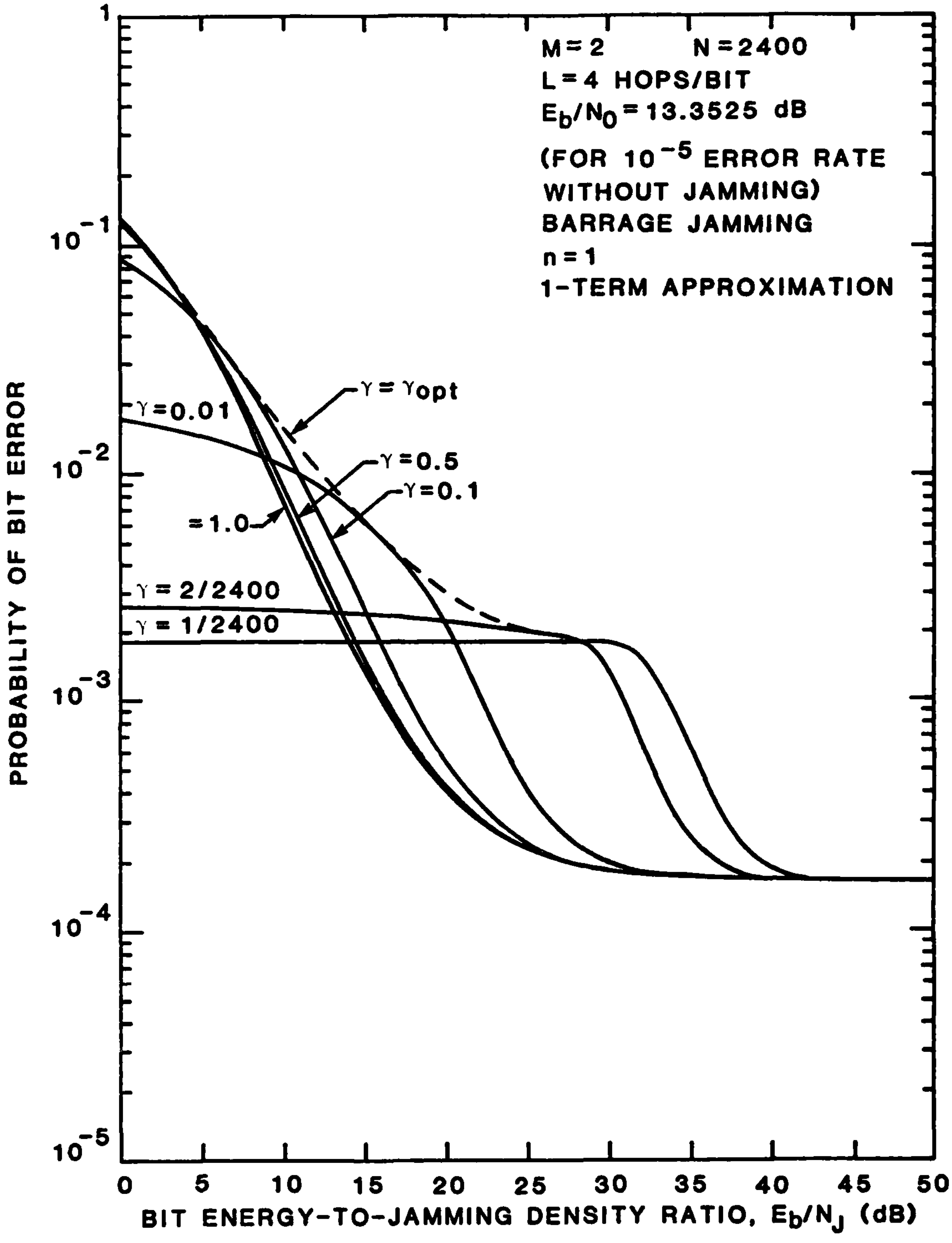


FIGURE 8-14 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n = 1) JAMMING WHEN L = 4 HOPS/BIT AND N = 2400 HOPPING SLOTS FOR BFSK/FH WITH E_b/N_0 = 13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

the straight-line behavior that is seen in the case of independent multi-tone jamming. This is due to the different values of $\Pi_L(\underline{\ell})$ for barrage jamming with n = 1, notably a significantly larger value of $\Pi_1(1, 1)$ and small values of $\Pi_1(1, 0)$ and $\Pi_1(0, 1)$.

Figure 8-15 summarizes the effects of barrage jamming with spacing n = 1 for several values of L. Again we see that increasing L cannot reduce the error rate if the jammer optimizes his fraction γ for the value of L the communicator uses.

Figures 8-16 through 8-18 show the effects of barrage jamming when the tone-spacing parameter is n = 1 for L = 1, L = 2, and L = 4 hops/bit, respectively. We note a dramatic difference between these figures and those for n = 1. Now the optimum-γ envelope again exhibits a linear behavior. This can be explained by the change in jamming event probabilities when the tones are located in every other cell, rather than contiguously. Now $\Pi_1(1, 1) = 0$ and $\Pi_1(0, 1)$ and $\Pi_1(1, 0)$ are the predominantly occurring jamming events.

For the spacing parameter n = 2 we have a minimum γ = 2/2400, since the spacing cannot be defined if there are fewer than two tones. Also, the maximum value of γ is now 0.5, since every other slot must be unjammed, by definition, when n = 2 for barrage jamming, and at most N/2 tones can be placed in the N hopping slots.

Figure 8-19 summarizes the effects of barrage jamming with n = 2. In this figure we plot the optimum-γ envelope with L = 1, L = 2, and L = 4 as a parameter. Again, we see that increasing L generally degrades

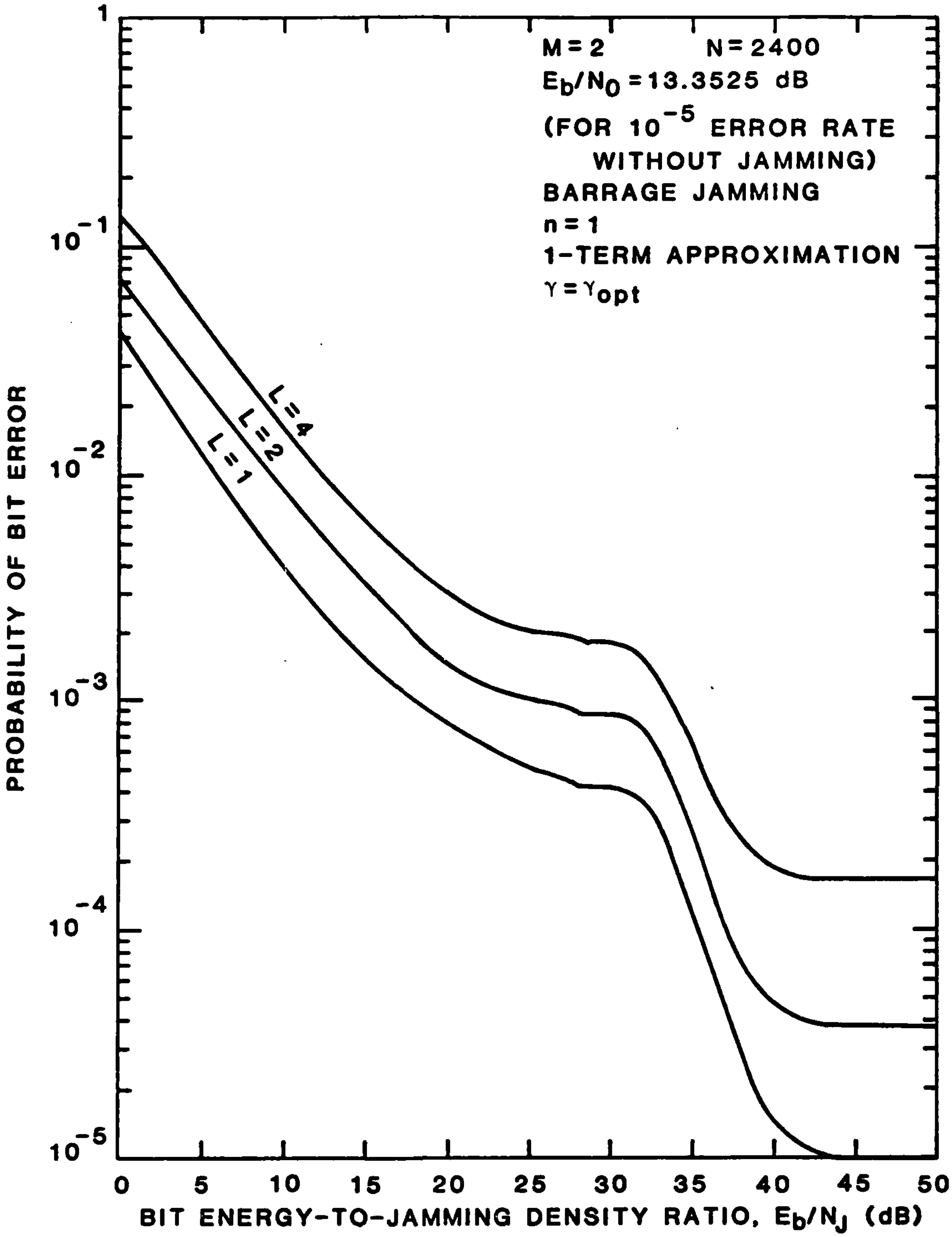


FIGURE 8-15 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH NUMBER OF HOPS PER BIT AS A PARAMETER FOR BARRAGE (n = 1) JAMMING WITH N = 2400 HOPPING SLOTS AND OPTIMUM JAMMING FRACTION FOR BFSK/FH WITH E_b/N_0 = 13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

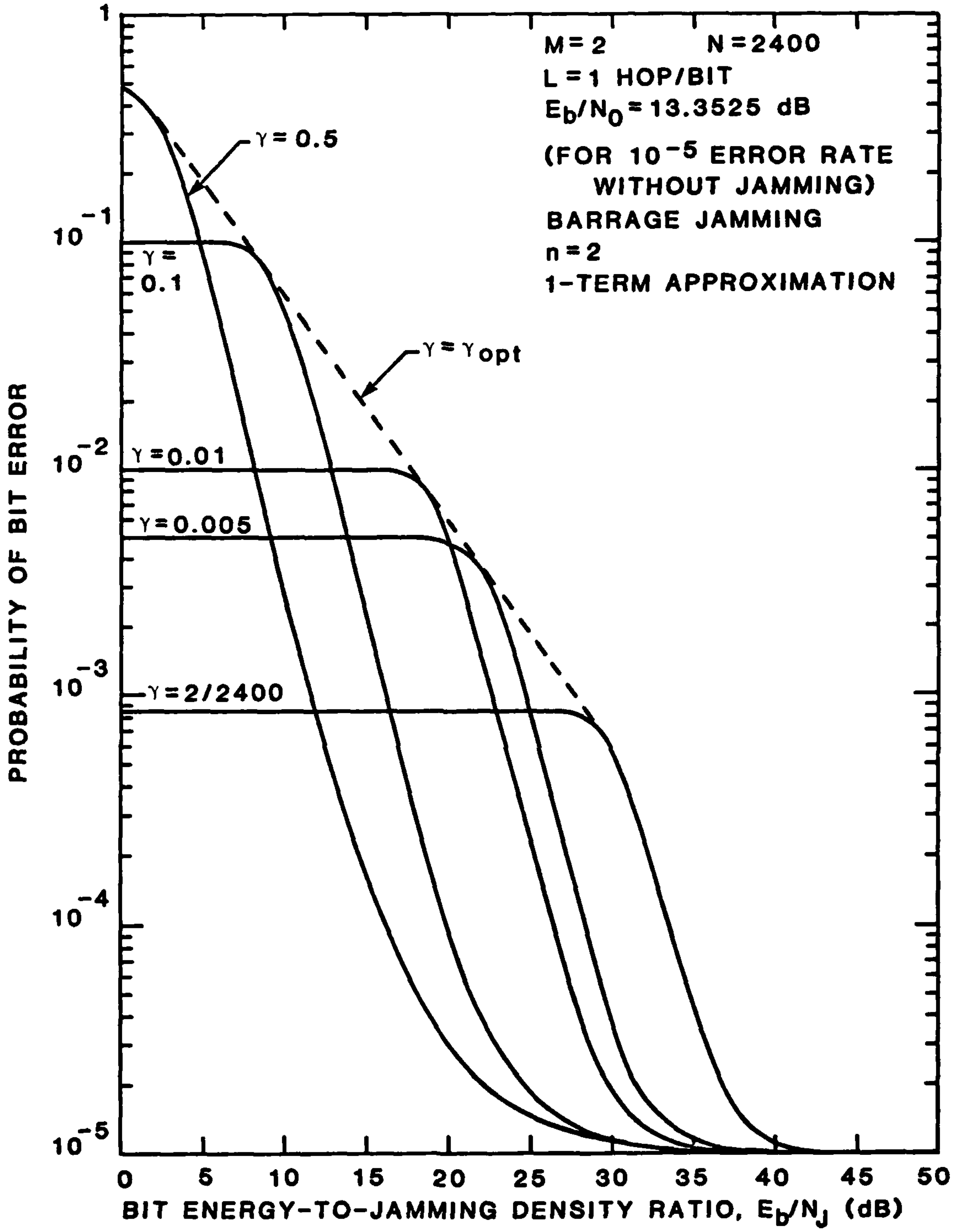


FIGURE 8-16 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE ($n=2$) JAMMING WHEN $L=1$ HOP/BIT AND $N=2400$ HOPPING SLOTS FOR BFSK/FH WITH $E_b/N_0=13.3525$ dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

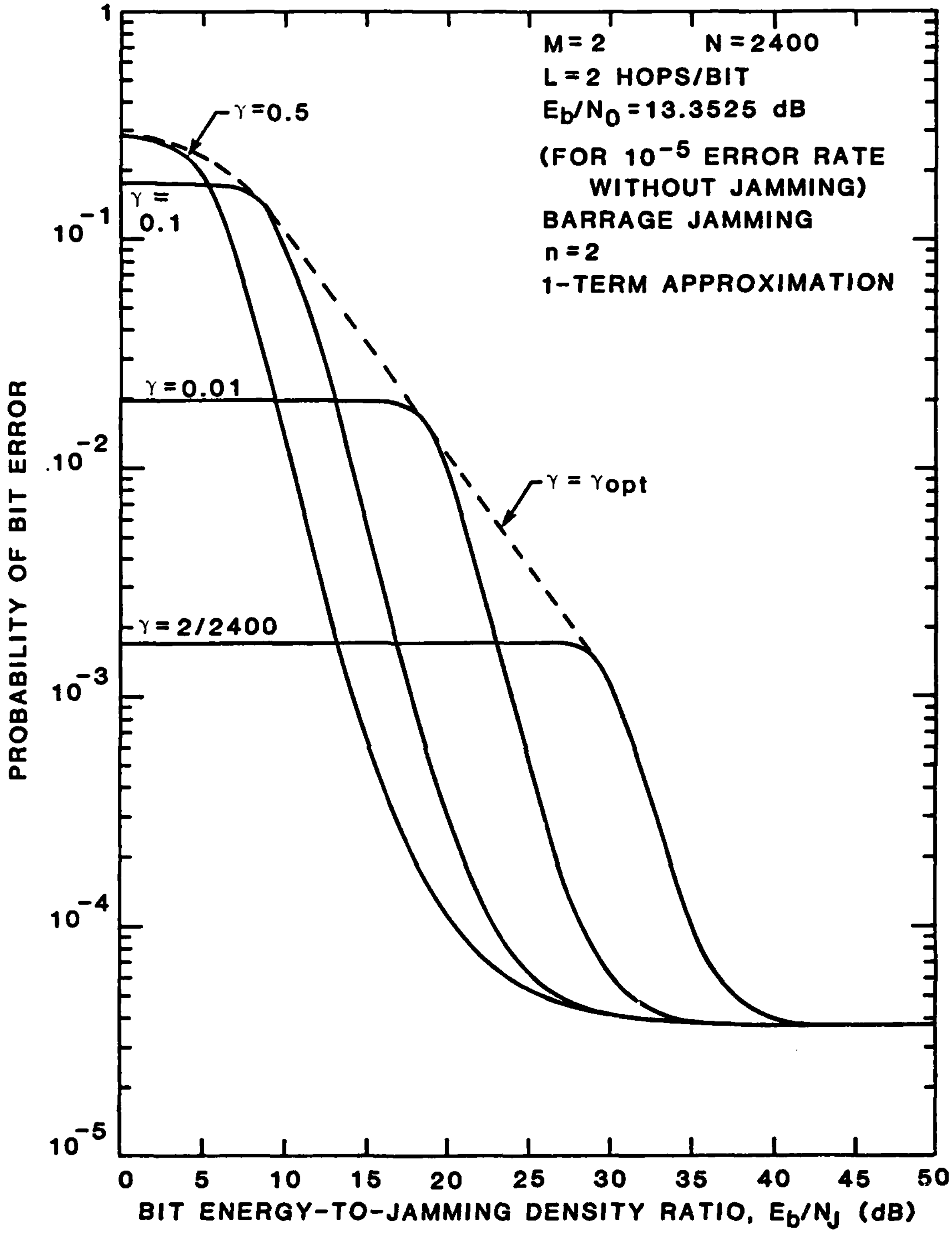
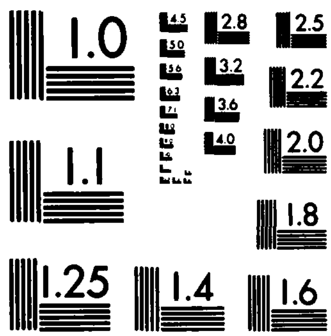


FIGURE 8-17 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n=2) JAMMING WHEN L=2 HOPS/BIT AND N=2400 HOPPING SLOTS FOR BFSK/FH WITH E_b/N_0=13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

AD-A147 766 OPTIMUM JAMMING EFFECTS ON FREQUENCY-HOPPING M-ARY FSK 5/7
(FREQUENCY-SHIFT K..(U) LEE (J S) ASSOCIATES INC
ARLINGTON VA J S LEE ET AL. OCT 84 JC-2025-N
UNCLASSIFIED N00014-83-C-0312 F/G 17/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

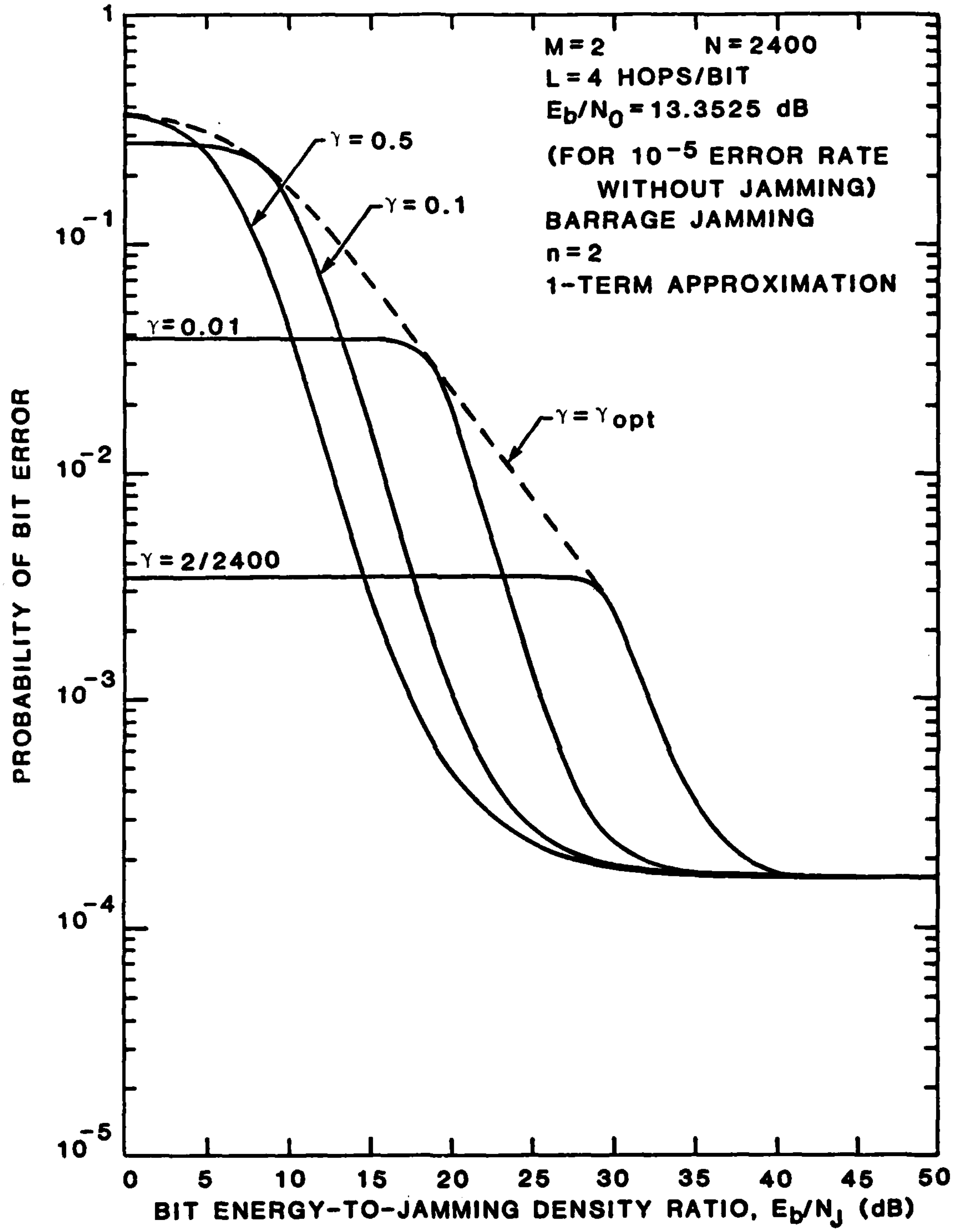


FIGURE 8-18 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n=2) JAMMING WHEN L=4 HOPS/BIT AND N=2400 HOPPING SLOTS FOR BFSK/FH WITH E_b/N_0 =13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

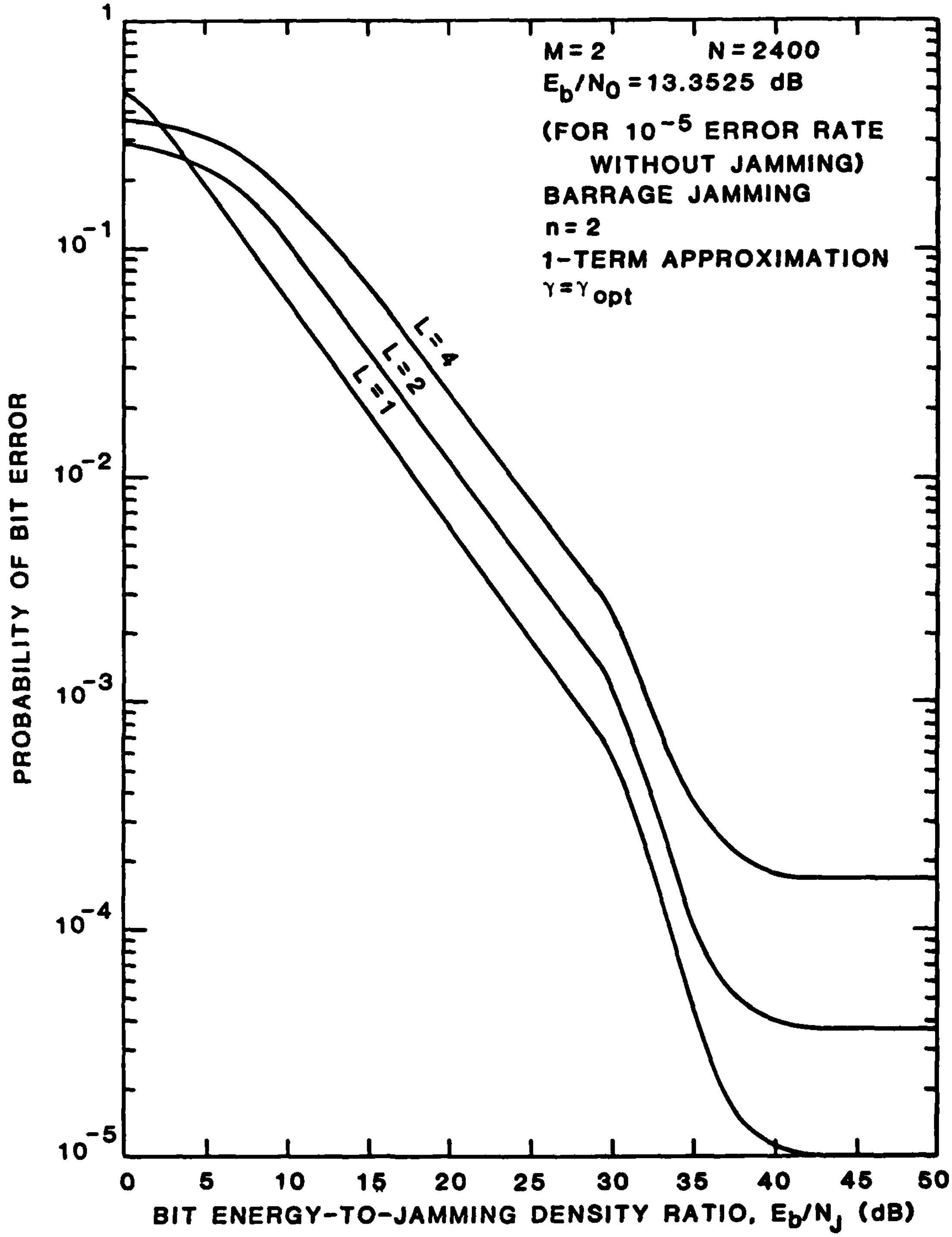


FIGURE 8-19 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH NUMBER OF HOPS PER BIT AS A PARAMETER FOR BARRAGE (n = 2) JAMMING WITH N = 2400 HOPPING SLOTS AND OPTIMUM JAMMING FRACTION FOR BFSK/FH WITH E_b/N_0 = 13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

the communicator's performance if the jammer can optimize the jamming strategy for the value of L selected by the communicator. However, for extremely strong jamming some improvement occurs for L = 2 hops/bit due to a limited form of quasi-diversity action. However when L is increased to 4, the combining loss increases and performance degrades relative to L = 2.

8.3.2.3 Effects of Band Multitone Jamming

For the band multitone jamming model, we again have two variations. As discussed in Section 8.2.3, we may have n = 1 tone per jammed signal band or n = 2 tones per signal band when M = 2. We consider each case separately.

Figures 8-20 through 8-22 show the effects of band multitone jamming with n = 1 tone per symbol band on BFSK/FH with jamming fraction γ as a parameter for L = 1, L = 2, and L = 4 hops/bit, respectively. As before, we also show the optimum-γ envelope. Here we again see the linear behavior of the optimum-γ curve. A summary curve, constructed as we did for the previous models, is shown in Figure 8-23. As was the case for barrage jamming with n = 2, we see a limited form of quasi-diversity improvement for very strong jamming, although noncoherent combining loss prevents increase of L from 2 to 4 from attaining a net improvement.

We note that for band multitone jamming with n = 1 the minimum realizable value of γ is γ = 1/2400 when just one tone is emitted by the jammer. The maximum value is γ = 0.5, since one filter in each possible hopping band remains unjammed.

The effects of band multitone jamming when there are n = 2 jamming tones per binary symbol band are shown in Figures 8-24 through 8-26 for L = 1, L = 2, and L = 4 hops/bit, respectively, with the fraction γ as a parameter. Here the minimum value of γ is 2/2400, since the model requires

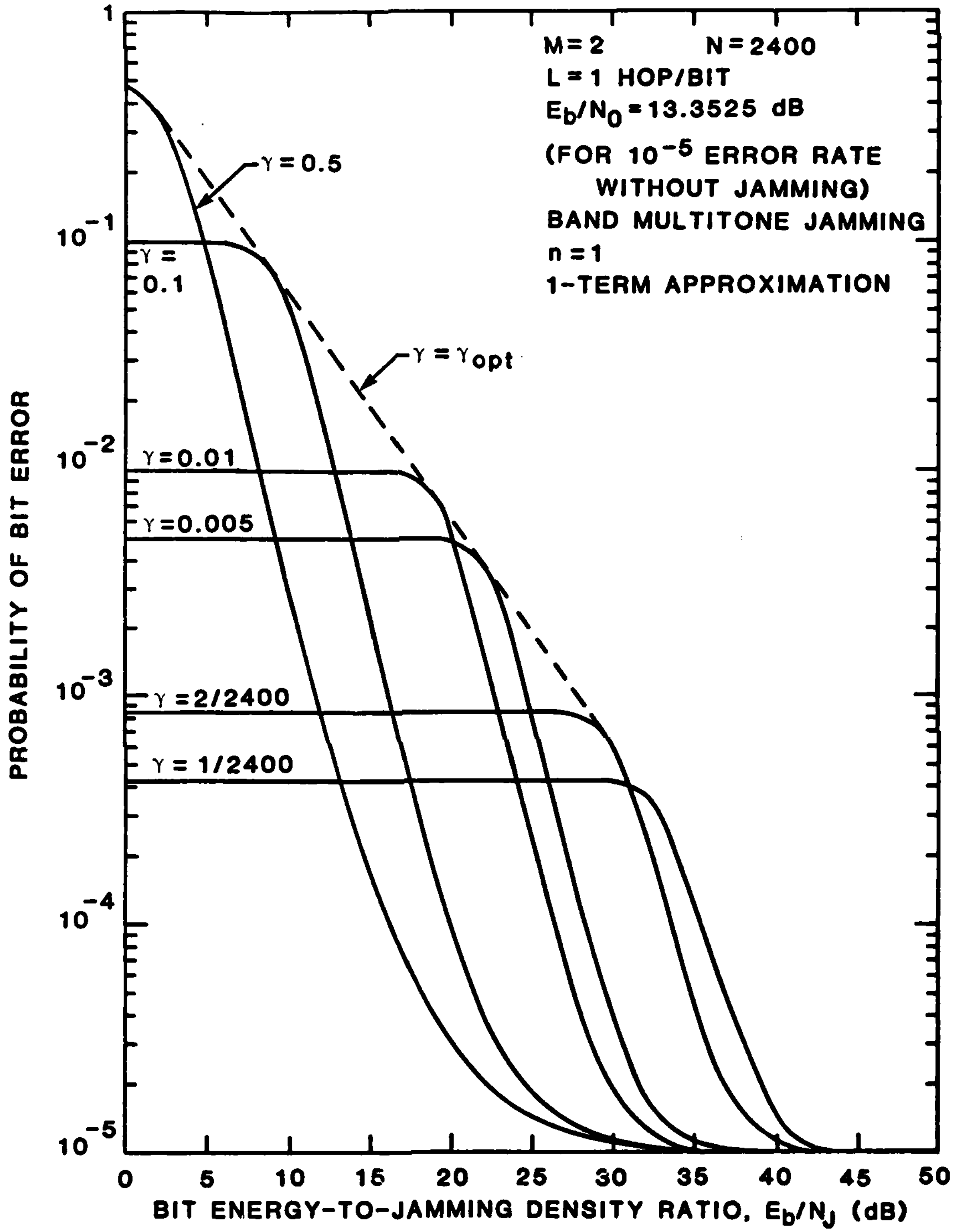


FIGURE 8-20 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BAND MULTITONE ($n=1$) JAMMING WHEN $L=1$ HOP/BIT AND $N=2400$ HOPPING SLOTS FOR BFSK/FH WITH $E_b/N_0=13.3525$ dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

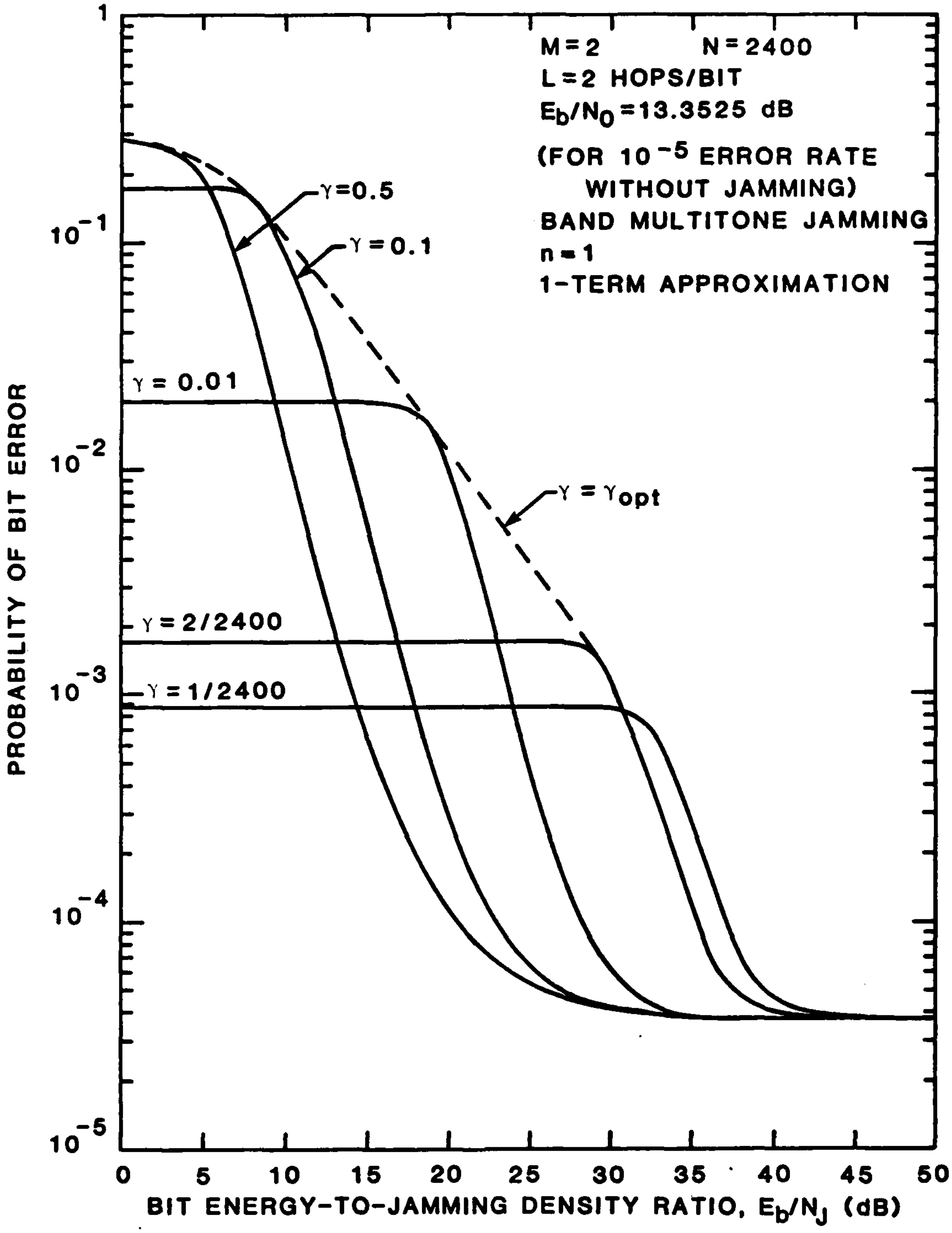


FIGURE 8-21 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BAND MULTITONE (n = 1) JAMMING WHEN L = 2 HOPS/BIT AND N = 2400 HOPPING SLOTS FOR BFSK/FH WITH E_b/N_0 = 13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

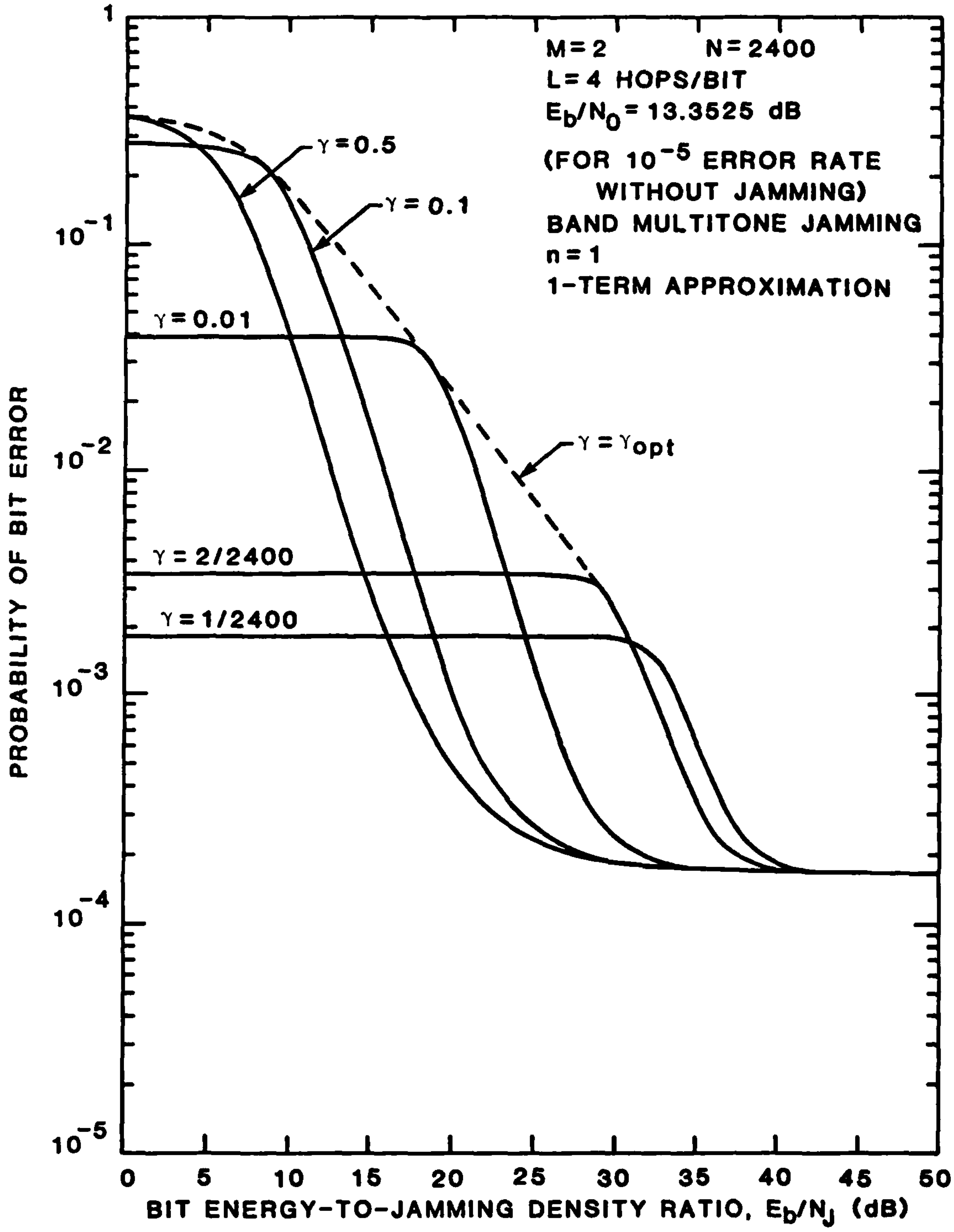


FIGURE 8-22 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BAND MULTITONE (n = 1) JAMMING WHEN L = 4 HOPS/BIT AND N = 2400 HOPPING SLOTS FOR BFSK/FH WITH E_b/N_0 = 13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

at least 2 jamming tones in one hopping band. The maximum value is $\gamma = 1.0$. Figure 8-27 summarizes the optimum-γ envelopes from Figures 8-24 through 8-26. Here we do not observe any quasi-diversity action for very low E_b/N_J. This is attributable to the difference in the system model required to make band multitone jamming realizable, namely nonoverlapping hopping positions, and to the lack of any jamming events under which the jamming tones might aid the correct decision (i.e., $\Pi_1(1, 0) = \Pi_1(0, 1) = 0$).

8.3.2.4 Summary of the Effects of the Different Jamming Models

In Figures 8-28 through 8-30 we collect together the optimum-γ envelopes for the five tone jamming models for L = 1, L = 2, and L = 4, respectively. We also have plotted the results for optimum partial-band noise jamming for comparison. We see from these curves that over much of the range of E_b/N_J the independent multitone (or randomly placed tones), barrage with spacing n = 2 slots, and band multitone with n = 1 tone per symbol band jammers are equally effective as the optimum jammer (from the jammer's viewpoint), regardless of L. The barrage jammer with n = 1 slot spacing and the band multitone jammer with n = 2 tones per symbol band are not effective jamming strategies.

We see that partial-band noise jamming, over most of the range of E_b/N_J, is not quite as effective as the optimum tone jamming strategies; however the difference in effectiveness is not so great as to rule out partial-band noise jamming if its hardware simplicity is desired. The apparent superiority of partial-band noise jamming at high values of E_b/N_J is a result of a slight difference in our analytical constraints for the case of partial-band noise jamming. When we determined the optimum jamming fraction for the case of partial-band noise jamming, we did not impose any

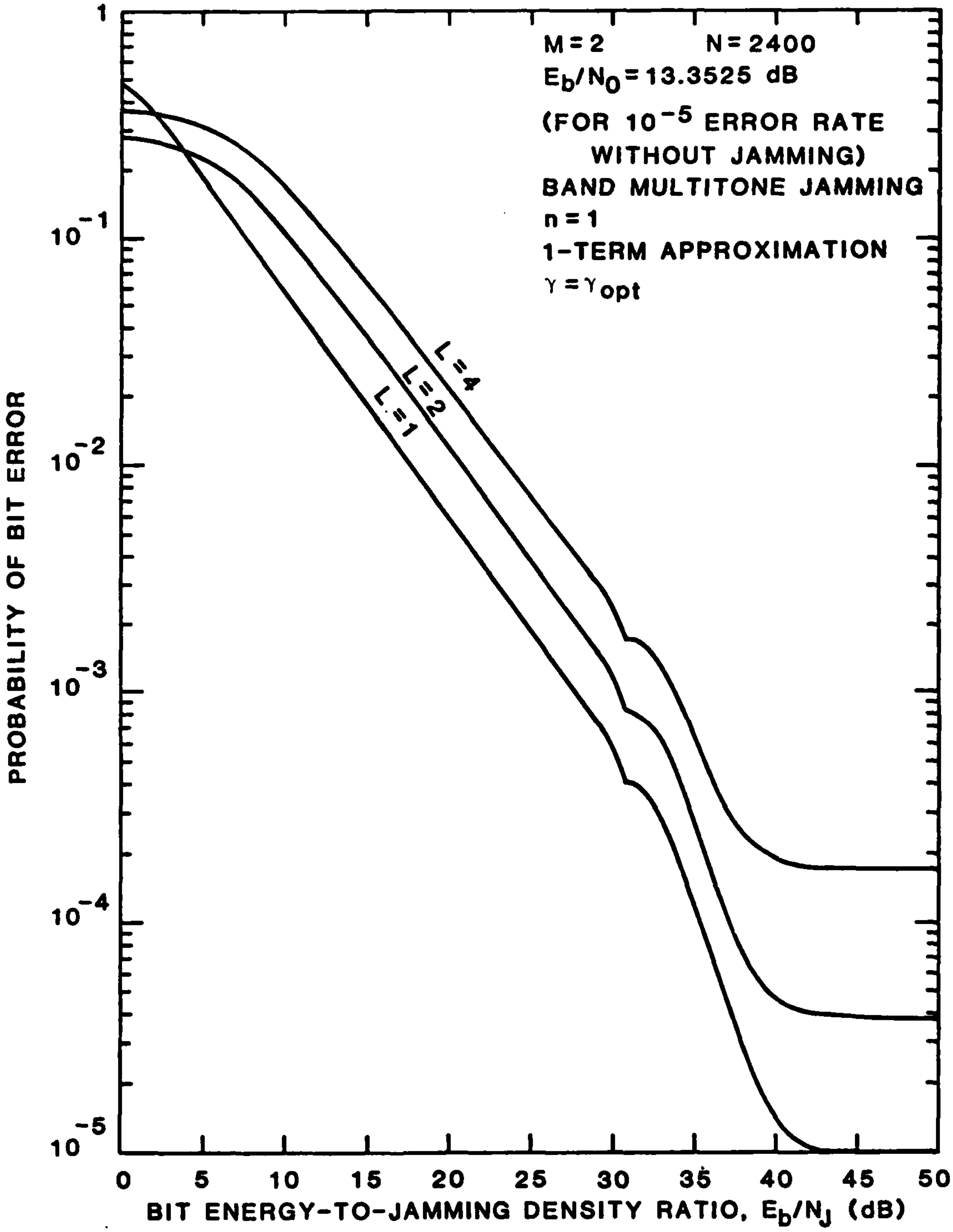


FIGURE 8-23 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH NUMBER OF HOPS PER BIT AS A PARAMETER FOR BAND MULTITONE (n = 1) JAMMING WITH N = 2400 HOPPING SLOTS AND OPTIMUM JAMMING FRACTION FOR BFSK/FH WITH E_b/N_0 = 13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

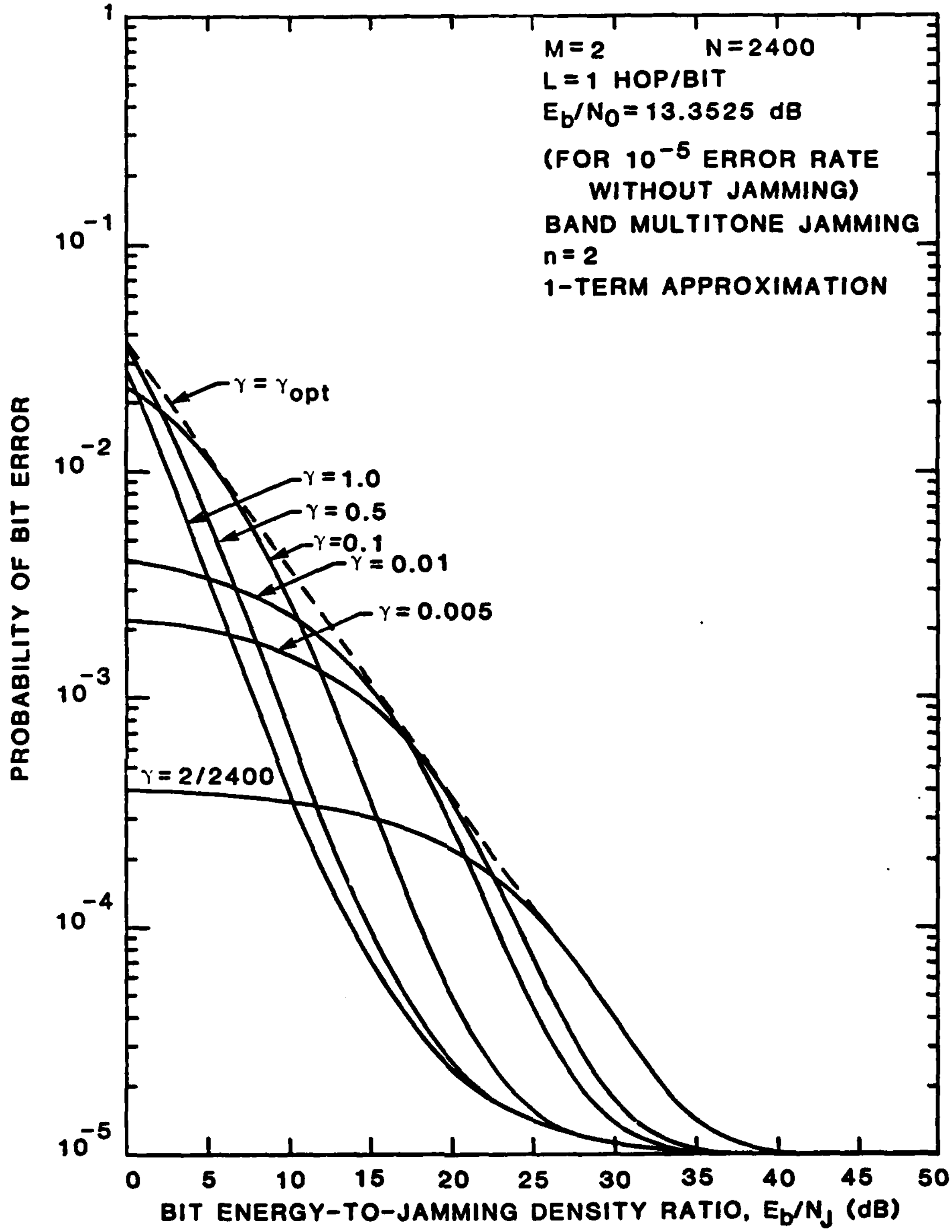


FIGURE 8-24 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BAND MULTITONE (n=2) JAMMING WHEN L=1 HOP/BIT AND N=2400 HOPPING SLOTS FOR BFSK/FH WITH E_b/N_0=13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

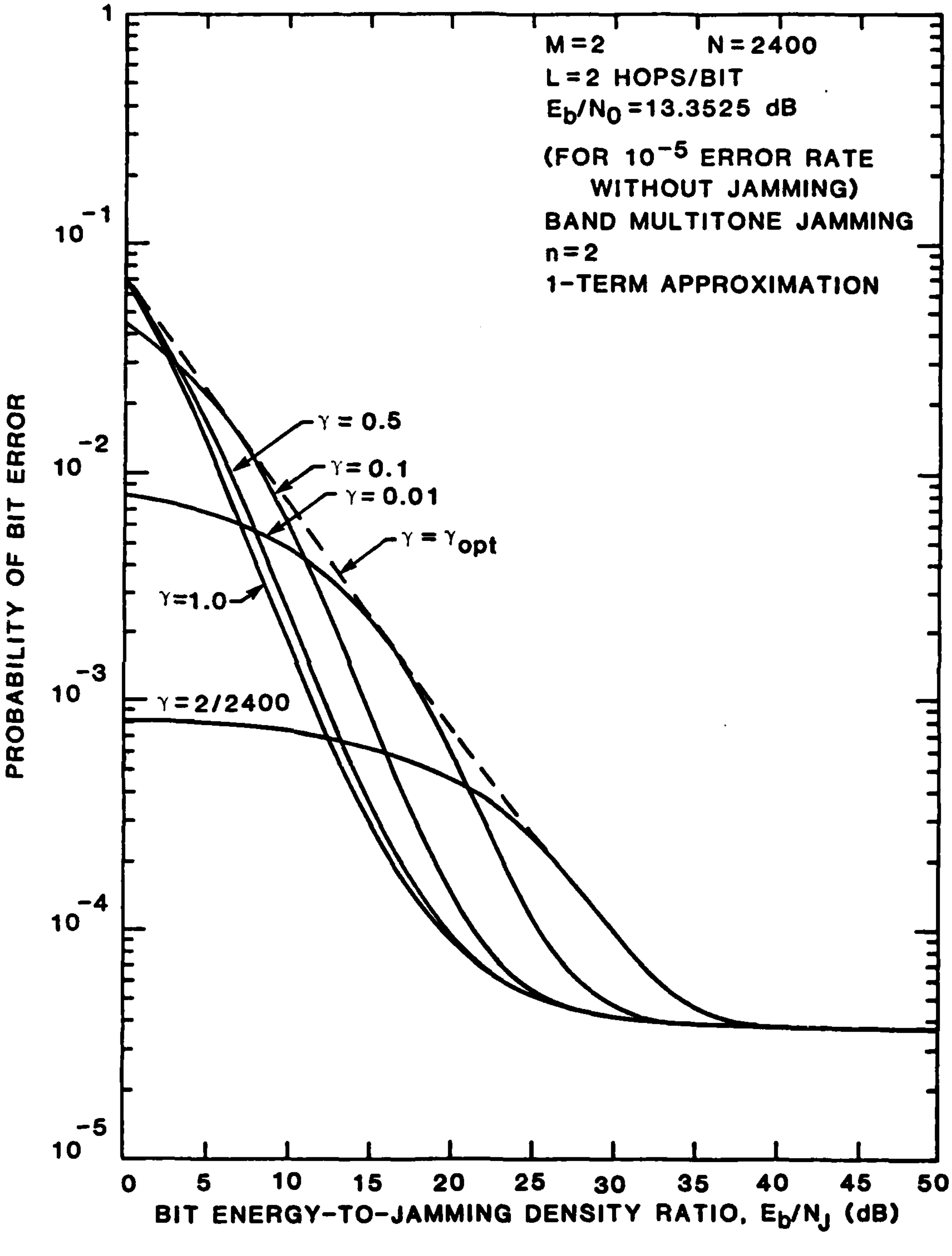


FIGURE 8-25 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BAND MULTITONE ($n=2$) JAMMING WHEN $L=2$ HOPS/BIT AND $N=2400$ HOPPING SLOTS FOR BFSK/FH WITH $E_b/N_0=13.3525$ dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

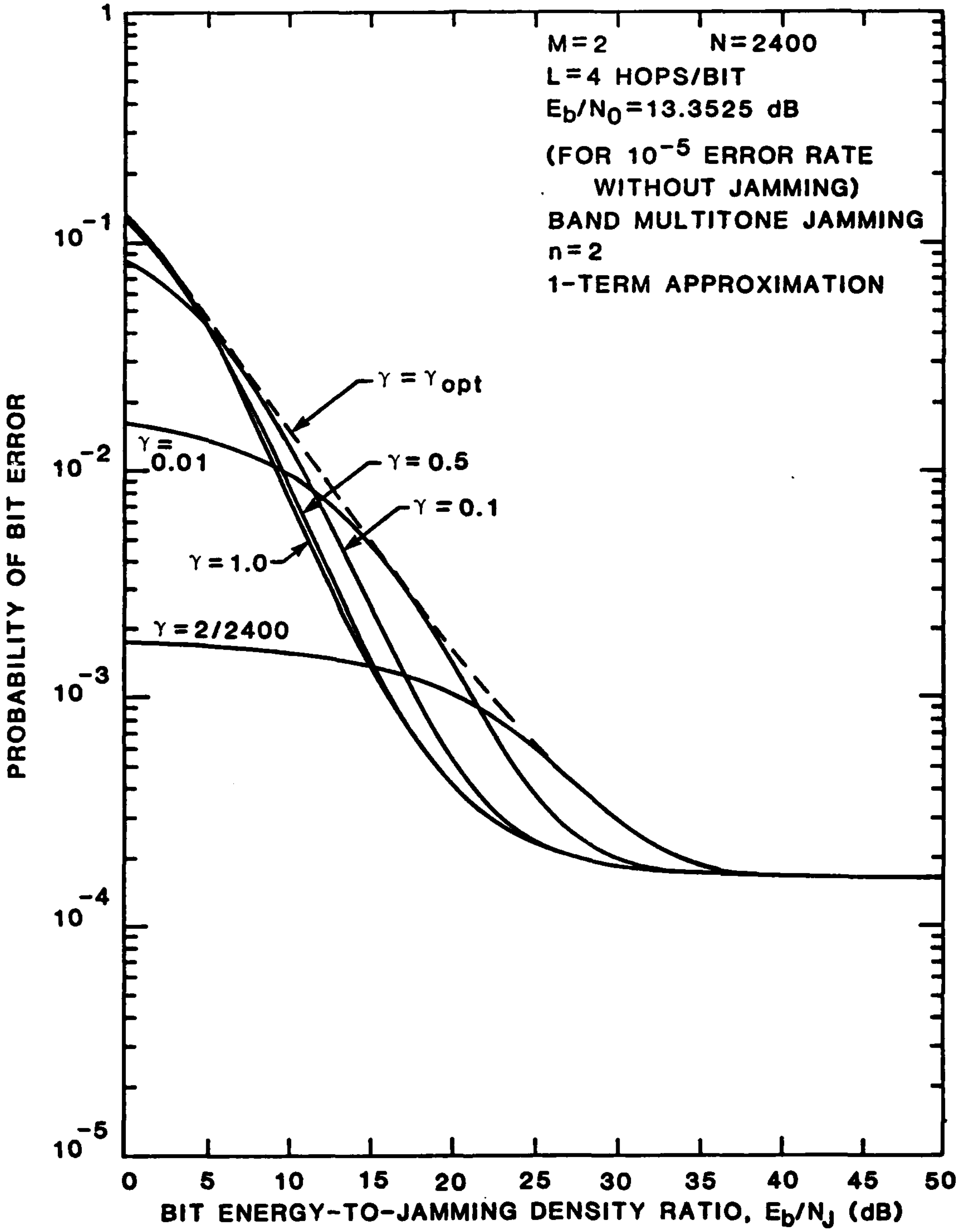


FIGURE 8-26 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BAND MULTITONE ($n=2$) JAMMING WHEN $L=4$ HOPS/BIT AND $N=2400$ HOPPING SLOTS FOR BFSK/FH WITH $E_b/N_0=13.3525$ dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

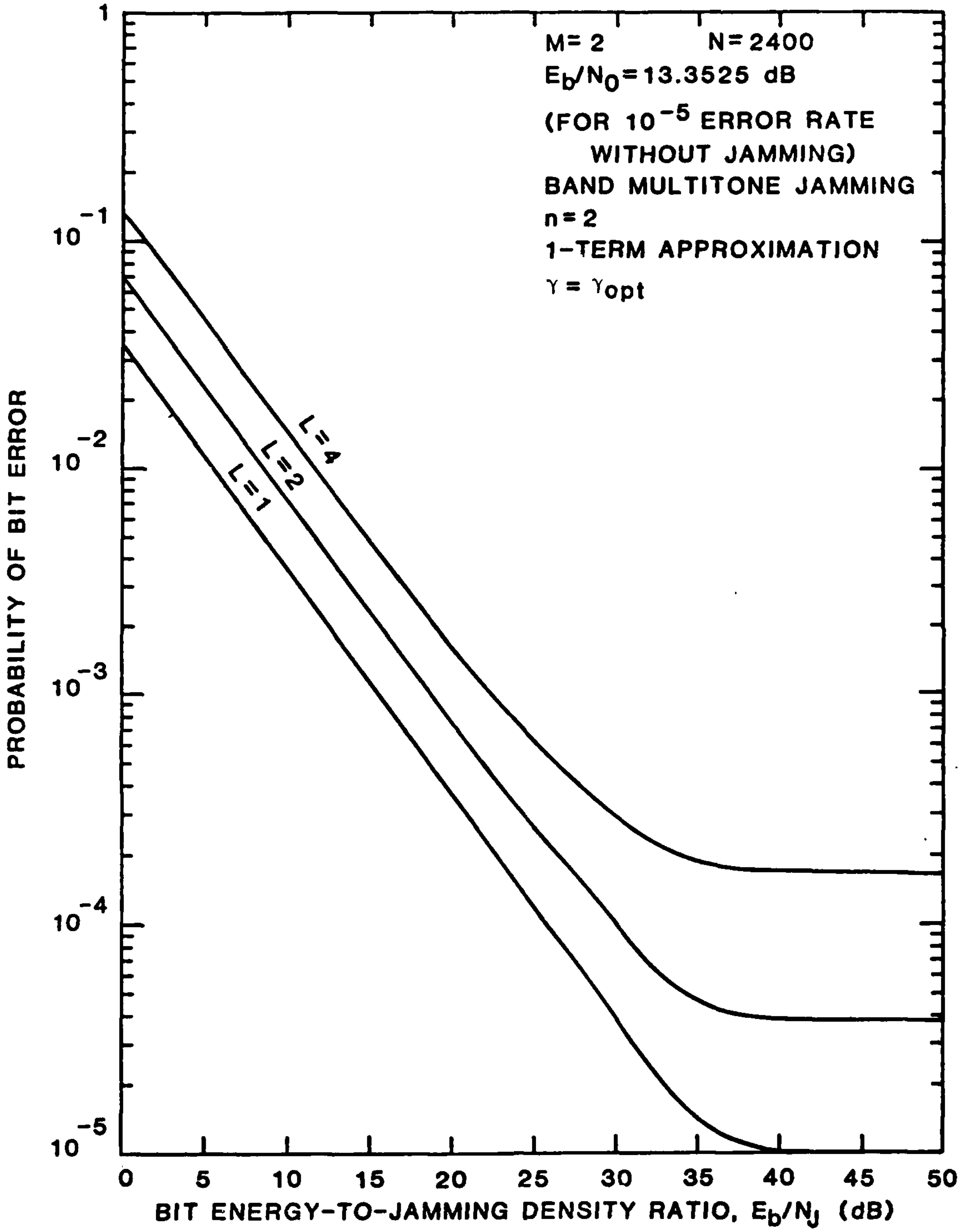


FIGURE 8-27 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH NUMBER OF HOPS PER BIT AS A PARAMETER FOR BAND MULTITONE (n=2) JAMMING WITH N=2400 HOPPING SLOTS AND OPTIMUM JAMMING FRACTION FOR BFSK/FH WITH E_b/N_0=13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

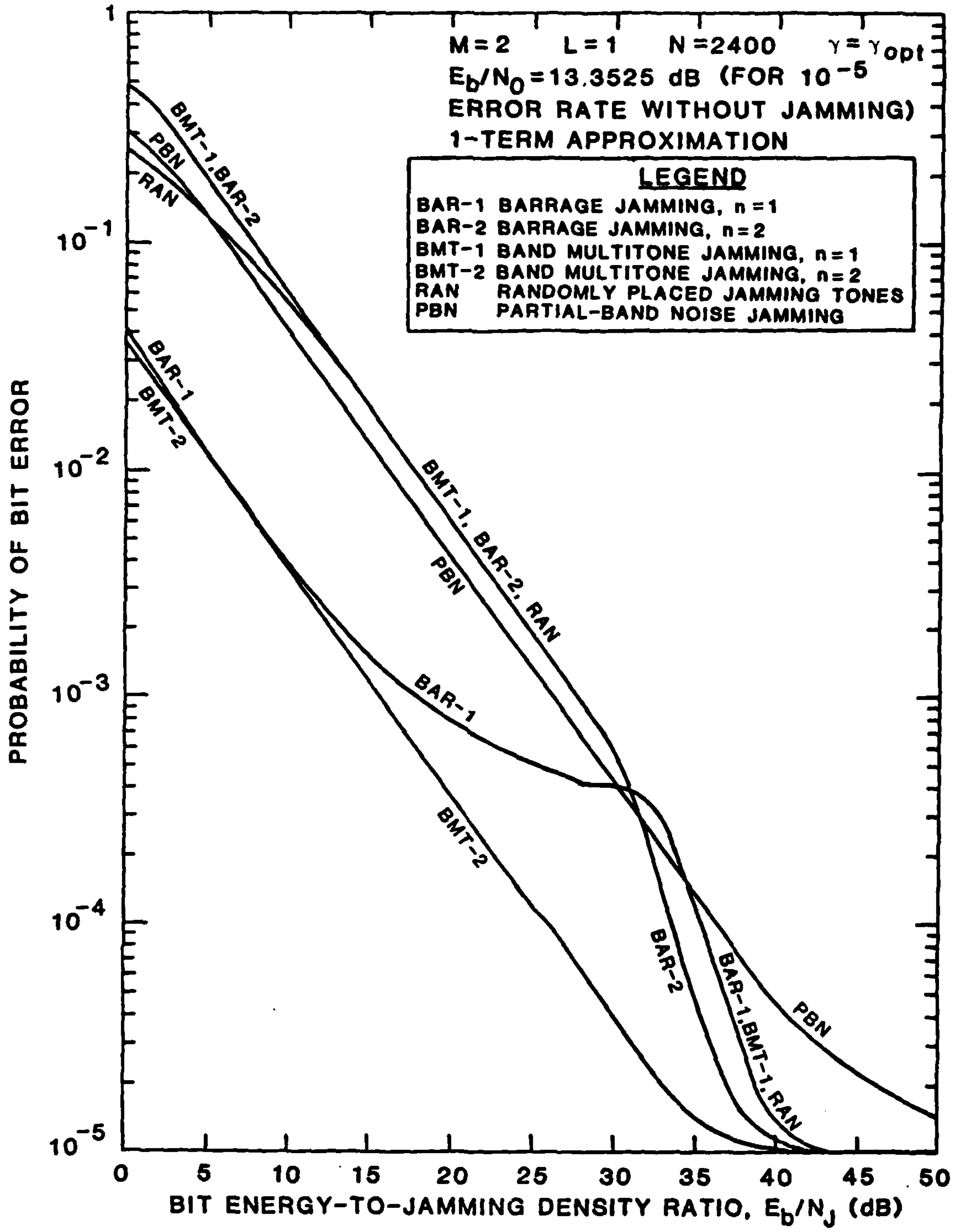


FIGURE 8-28 COMPARISON OF JAMMING STRATEGIES AGAINST SQUARE-LAW LINEAR COMBINING RECEIVER FOR BFSK/FH WITH L=1 HOP/BIT, N=2400 HOPPING SLOTS, AND $E_b/N_0 = 13.3525$ dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

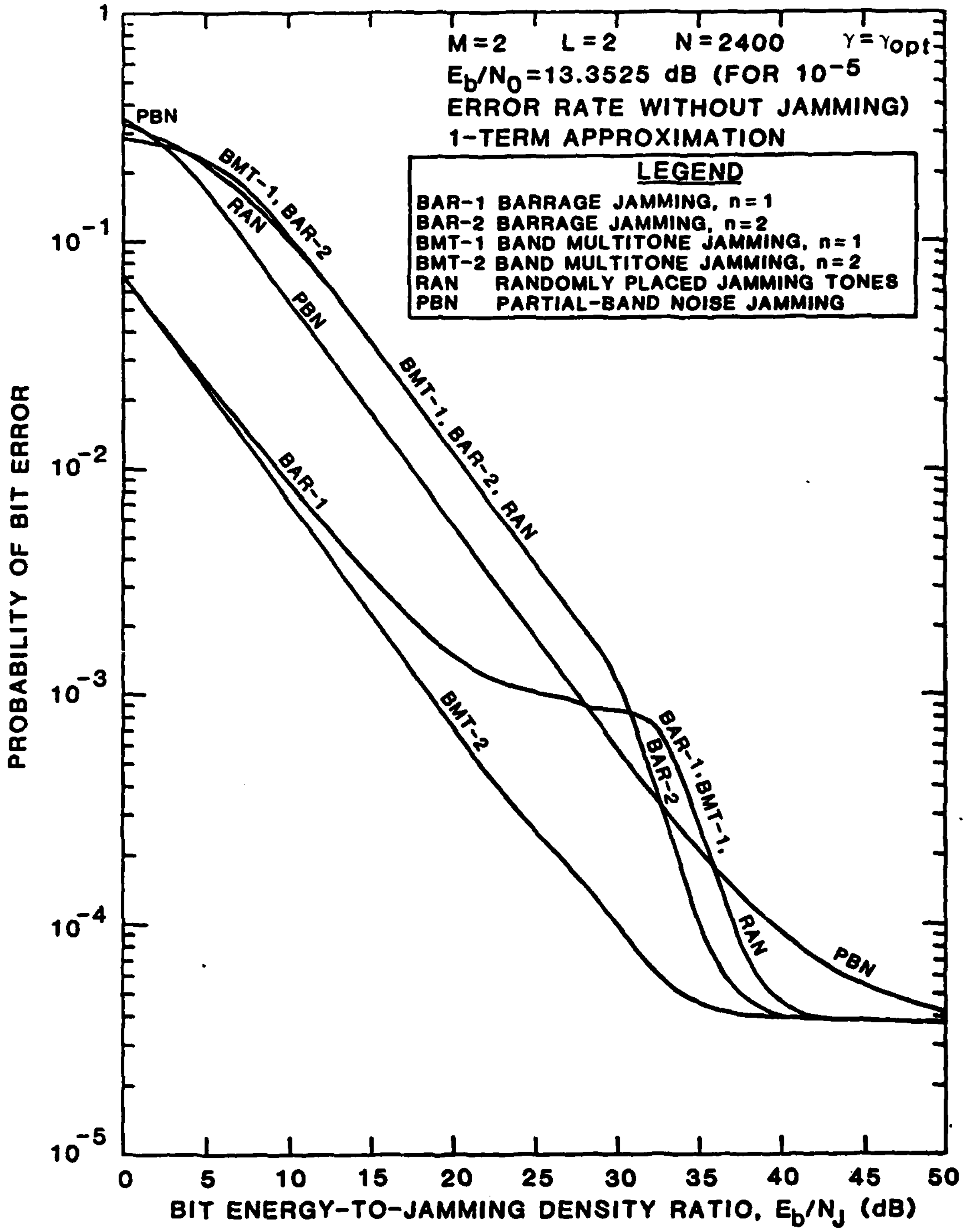


FIGURE 8-29 COMPARISON OF JAMMING STRATEGIES AGAINST SQUARE-LAW LINEAR COMBINING RECEIVER FOR BFSK/FH WITH L=2 HOPS/BIT, N=2400 HOPPING SLOTS, AND E_b/N_0=13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

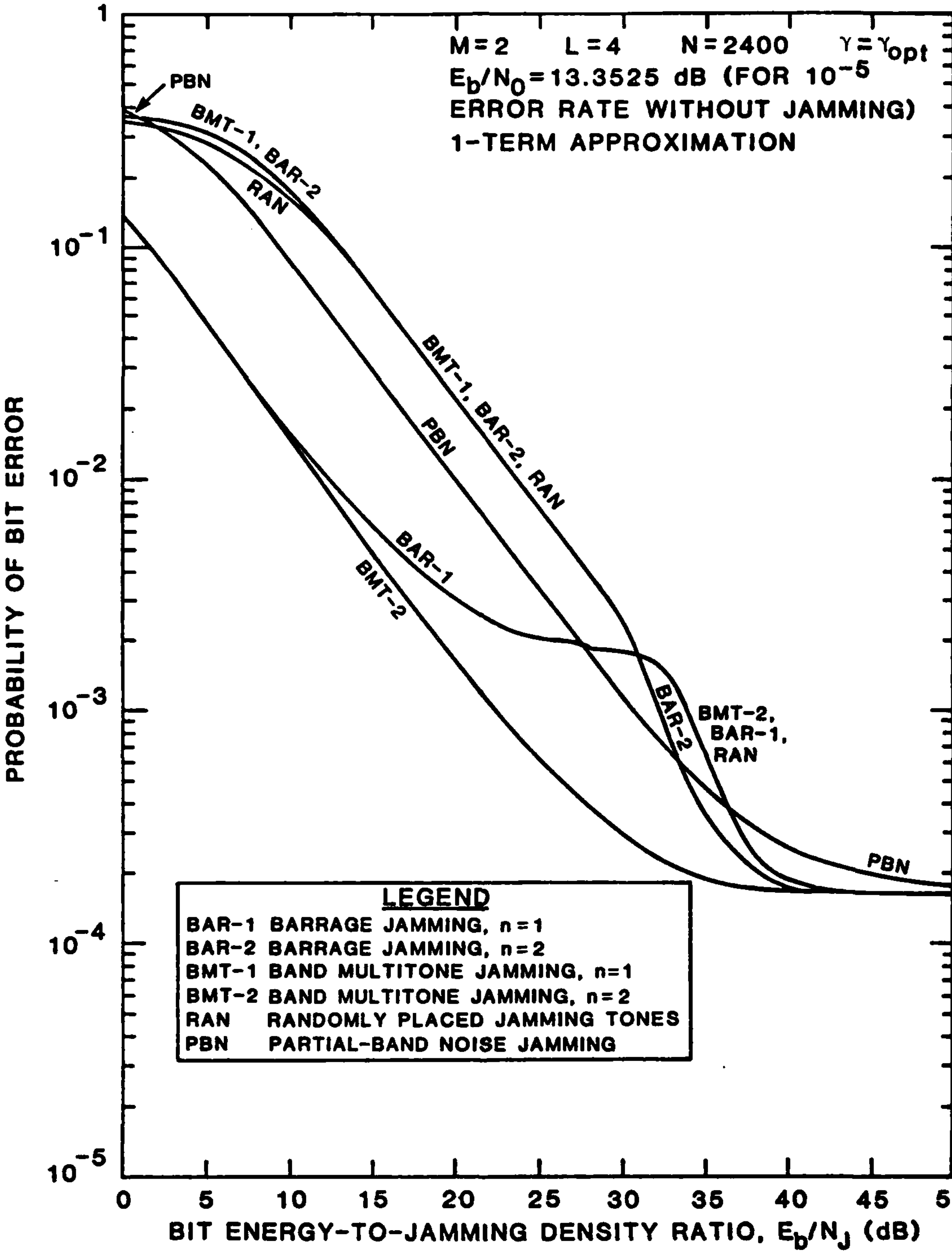


FIGURE 8-30 COMPARISON OF JAMMING STRATEGIES AGAINST SQUARE-LAW LINEAR COMBINING RECEIVER FOR BFSK/FH WITH $L=4$ HOPS/BIT, $N=2400$ HOPPING SLOTS, AND $E_b/N_0=13.3525$ dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

constraint on how small γ was allowed to become. If we refer back to Figure 2-26, for example, we see that for E_b/N_J = 30 dB that $\gamma_0 \approx 4.5 \times 10^{-4}$; for higher values of E_b/N_J our optimization procedure allowed γ_0 to approach arbitrarily close to zero. If we were to impose a constraint that $\gamma \geq MB/W = M/N$ on the partial-band noise optimization process, we would find that the curves would fall off rapidly, as do the tone jamming curves, when E_b/N_J increases and the apparent superiority of partial-band noise jamming in this region is likely to disappear.

In conclusion, we have found that the optimum tone jamming strategy against an L-hops/symbol MFSK/FH system is either barrage jamming with spacing between tones of n = M, or the essentially equivalent form of band multitone jamming with n = 1 jamming tone per M-ary symbol band.

8.3.3 Results for Worst-Case Tone Jamming

Based on the approximate error rate equation given by (8-46)-(8-48), we have identified the worst-case (from the communicator's viewpoint) tone jamming against MFSK/FH as barrage jamming with the tone spacing n = M or band multitone jamming with n = 1 tone per M-ary symbol band. We now concentrate our attention on this worst-case jamming using the barrage (n = M) model for further analysis using the exact form of the error rate equations from (8-18) and (8-19). We also, for the sake of computational efficiency, have made use of the special-case equations (8-41)-(8-45) when L = 1 hop/symbol. The computer programs used are given in Appendix 8G (special-case equations) and Appendix 8H (general form).

As a typical example of a practical MFSK/FH system, we have selected a system using N = 2400 hopping frequencies. We have also chosen to use those values of E_b/N_0 for which an ideal MFSK system will achieve $P_b(e) = 10^{-5}$

in the absence of jamming. These values of E_b/N_0 are summarized in Table 8-3 for the values of M we have considered.

With regard to the jamming fraction $\gamma = q/N$, we have computed results for 7 values of γ. Six values, as summarized in Table 8-4, are common to all values of M considered. The seventh value is $\gamma = 1/M$, which is the maximum realizable value for barrage jamming with tone spacing $n = M$. The three smallest values of γ show the effects of the discrete nature of tone jamming on the curve of optimum jamming performance, while the remaining values permit us to draw a smoothed approximation to the optimum jamming curve without the necessity of computing 2400/M individual curves.

Figures 8-31 through 8-33 plot the numerical results obtained for M = 2 and L = 1, L= 2, and L = 3 hops/symbol, respectively. Figure 8-34 summarizes the results for M = 2 by combining in one graph the worst case (jammer's optimum) performance curves from Figures 8-31 through 8-32. We observe from these curves that the jammer must choose the correct number of jamming tones carefully, for an incorrect choice (i.e. $\gamma \neq \gamma_{opt}$) may result in a lessening of the jamming effectiveness by more than an order of magnitude. The importance of the correct choice of the number of tones is especially important when the optimum number is small. For example, from Figure 8-32 we see that incorrectly choosing 12 tones instead of 3 tones when E_b/N_J = 27.5 dB degrades the jamming effectiveness by nearly an order of magnitude--communicator's BER under the choice of 3 jamming tones is 2.1×10^{-3}, whereas if the jammer uses 12 tones the communicator's BER is only 3.8×10^{-4}.

From the summary curves in Figures 8-34, we observe that over most of the range of E_b/N_J, there is a degradation of performance as L increases. This is due to the noncoherent combining loss (see Section 2.1.2 for dis-

TABLE 8-3

BIT ENERGY-TO-THERMAL NOISE RATIOS FOR WHICH IDEAL MFSK HAS $P_b(e) = 10^{-5}$ IN THE ABSENCE OF JAMMING

M	E_b/N_0
2	13.3525 dB
4	10.6065 dB
8	9.0939 dB

TABLE 8-4

SUMMARY OF PARTIAL-BAND JAMMING FRACTIONS
USED IN NUMERICAL COMPUTATIONS

FRACTION γ	NUMBER OF JAMMING TONES q
$1/2400 \simeq 4.16667 \times 10^{-4}$	1
$2/2400 \simeq 8.33333 \times 10^{-4}$	2
$3/2400 = 0.00125$	3
0.005	12
0.01	24
0.1	240
1/M	2400/M

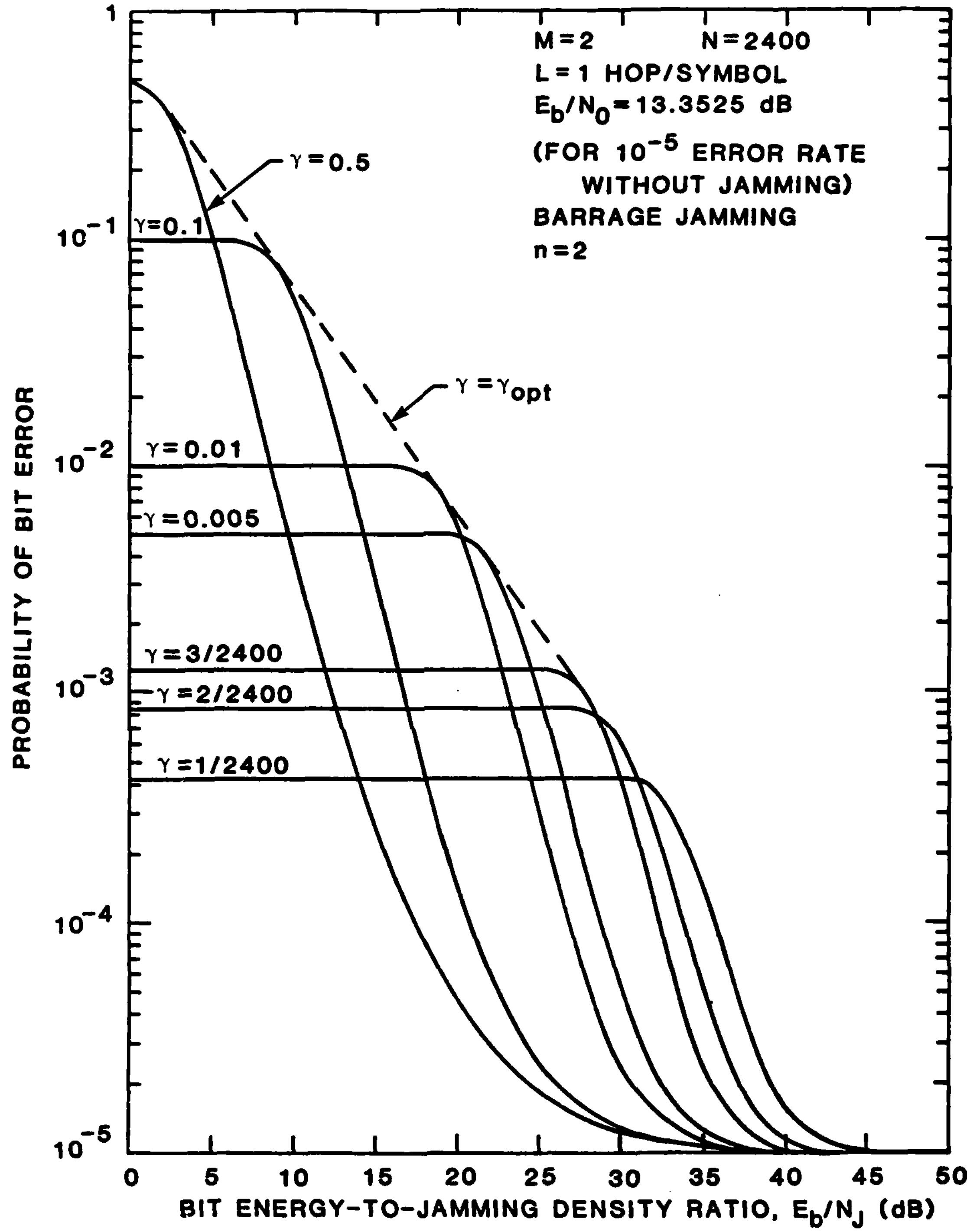


FIGURE 8-31 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n=2) JAMMING AGAINST MFSK/FH FOR M=2 WITH L=1 HOP/SYMBOL, N=2400 HOPPING SLOTS, AND E_b/N_0=13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

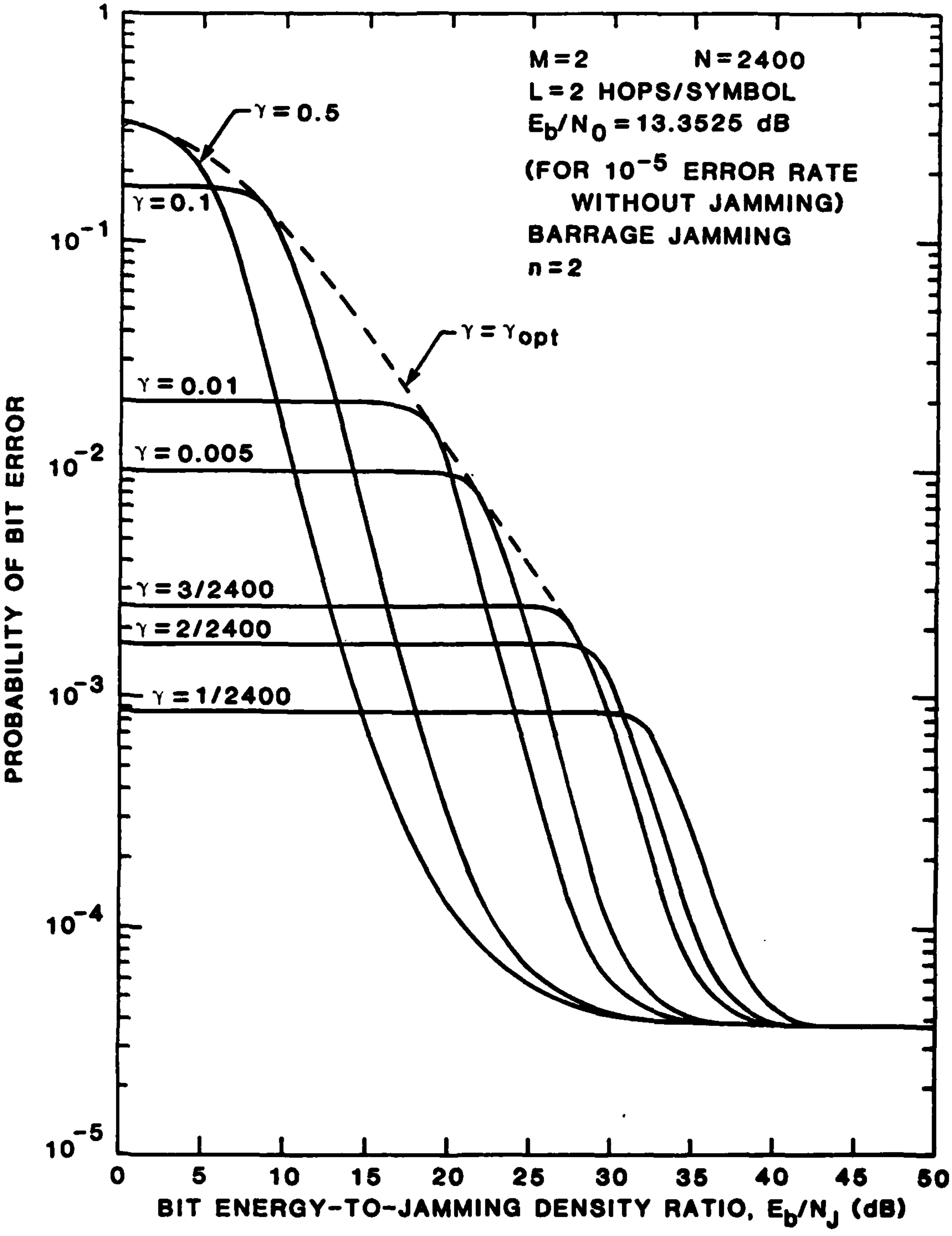


FIGURE 8-32 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n=2) JAMMING AGAINST MFSK/FH FOR M=2 WITH L=2 HOPS/SYMBOL, N=2400 HOPPING SLOTS, AND E_b/N_0=13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

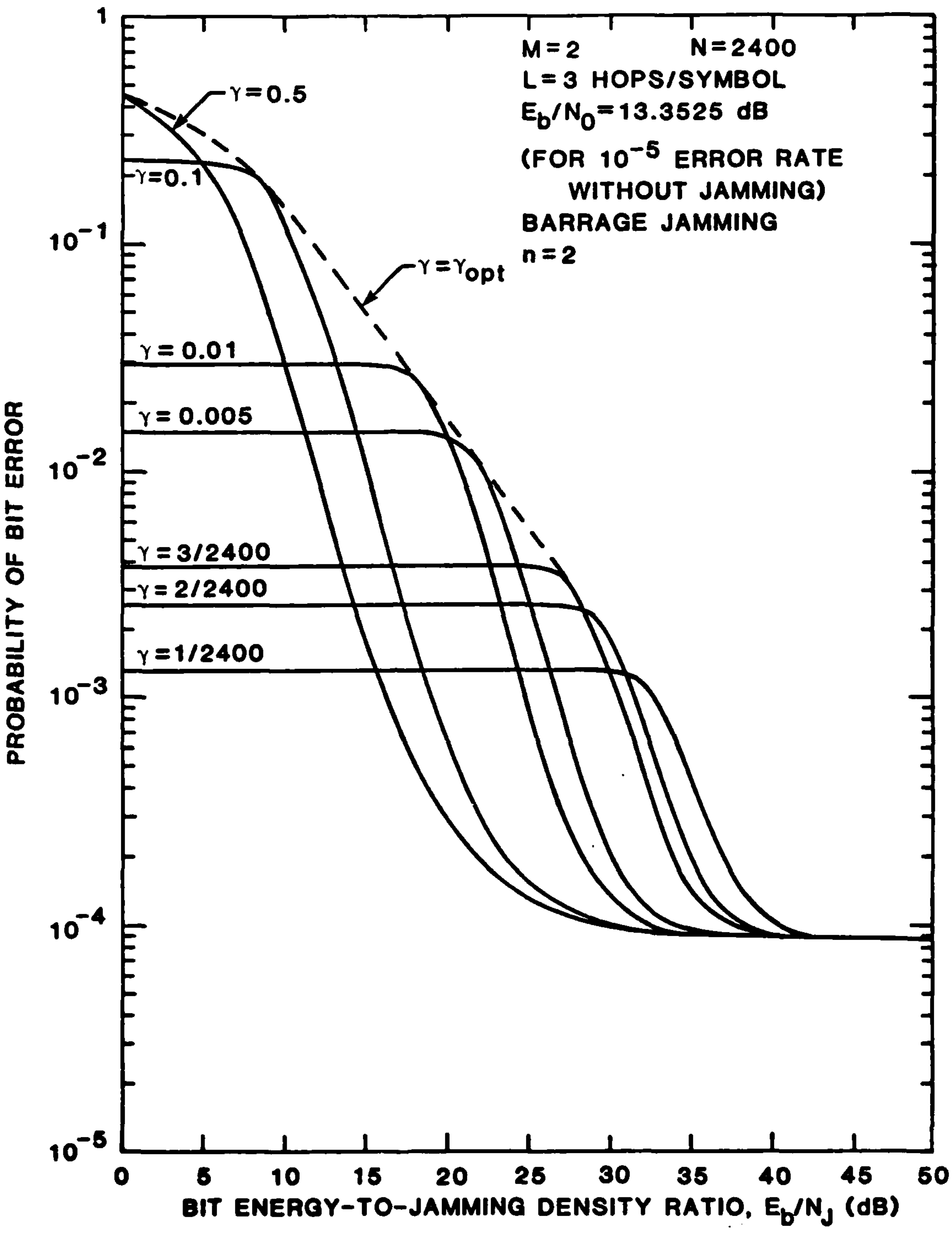


FIGURE 8-33 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n=2) JAMMING AGAINST MFSK/FH FOR M=2 WITH L=3 HOPS/SYMBOL, N=2400 HOPPING SLOTS, AND E_b/N_0=13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

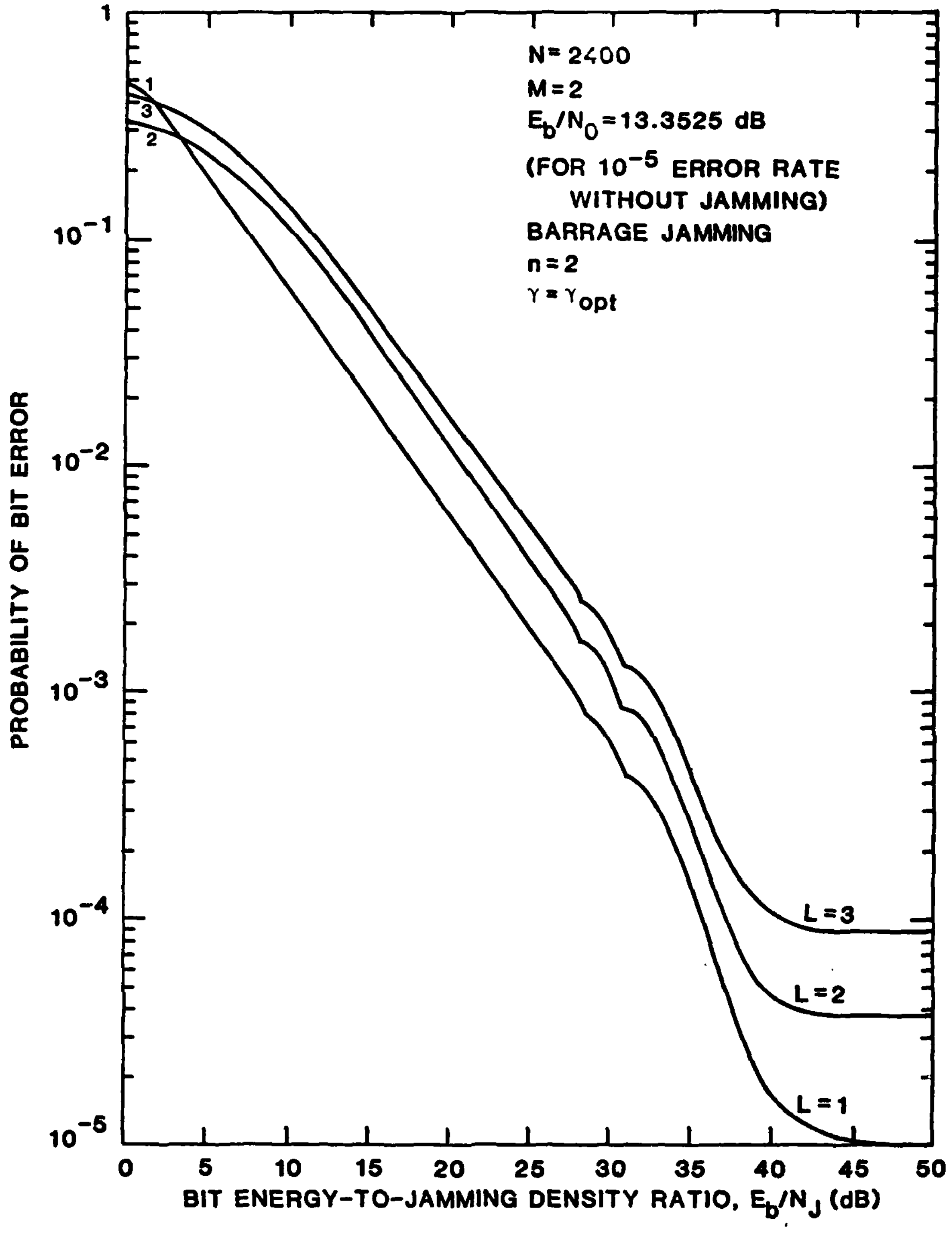


FIGURE 8-34 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH NUMBER OF HOPS PER SYMBOL AS A PARAMETER FOR BARRAGE (n=2) JAMMING AGAINST MFSK/FH FOR M=2 WITH OPTIMUM JAMMING FRACTION, N=2400 HOPPING SLOTS, AND E_b/N_0=13.3525 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

cussion of this topic). However, for E_b/N_0 less than about 3.2 dB, a limited amount of a quasi-diversity improvement is noted when L increases from 1 to 2. The limited amount of improvement thus available is clearly shown by the fact that increasing L from 2 to 3 does not yield further improvement, but rather degrades the performance. In any event, this area is of little practical significance, since the bit error probability exceeds 0.25 regardless of the number of hops per symbol.

Figures 8-35 through 8-37 show the performance of the MFSK/FH system when M = 4 with L = 1, L = 2, and L = 3 hops/symbol, respectively. The optimum jamming curves for M = 4 are summarized in Figure 8-38. Again we see that the noncoherent combining loss prevents any realization of any quasi-diversity gain, except at very low E_b/N_J, less than about 3 dB. In the case of M = 4, we note that there is a region where L = 3 hops/symbol performs marginally better (about 3% lower BER) than L = 2, but this is in the region where $P_b(e) \simeq 0.4$.

Figures 8-39 and 8-40 show the performance of the MFSK/FH system when M = 8 with L = 1 and L = 2 hops/symbol, respectively. Because of the rapid growth of computational time with M and L, we have omitted the case L = 3, M = 8 from our numerical results. Figure 8-41 summarizes the optimum jamming curves for the two values of L. Again, we see similar behavior with higher values of L giving an advantage only for low E_b/N_J.

Finally, Figures 8-42 and 8-43 compare the performance of the tone jamming for different values of M when L = 1 and L = 2 hops/symbol, respectively. The figures also show in dashed lines the corresponding performance curves for optimum partial-band noise jamming. We observe that over most of the range of E_b/N_J the barrage tone jamming with tone spacing *n = M frequency cells is a more effective jamming strategy than partial-*

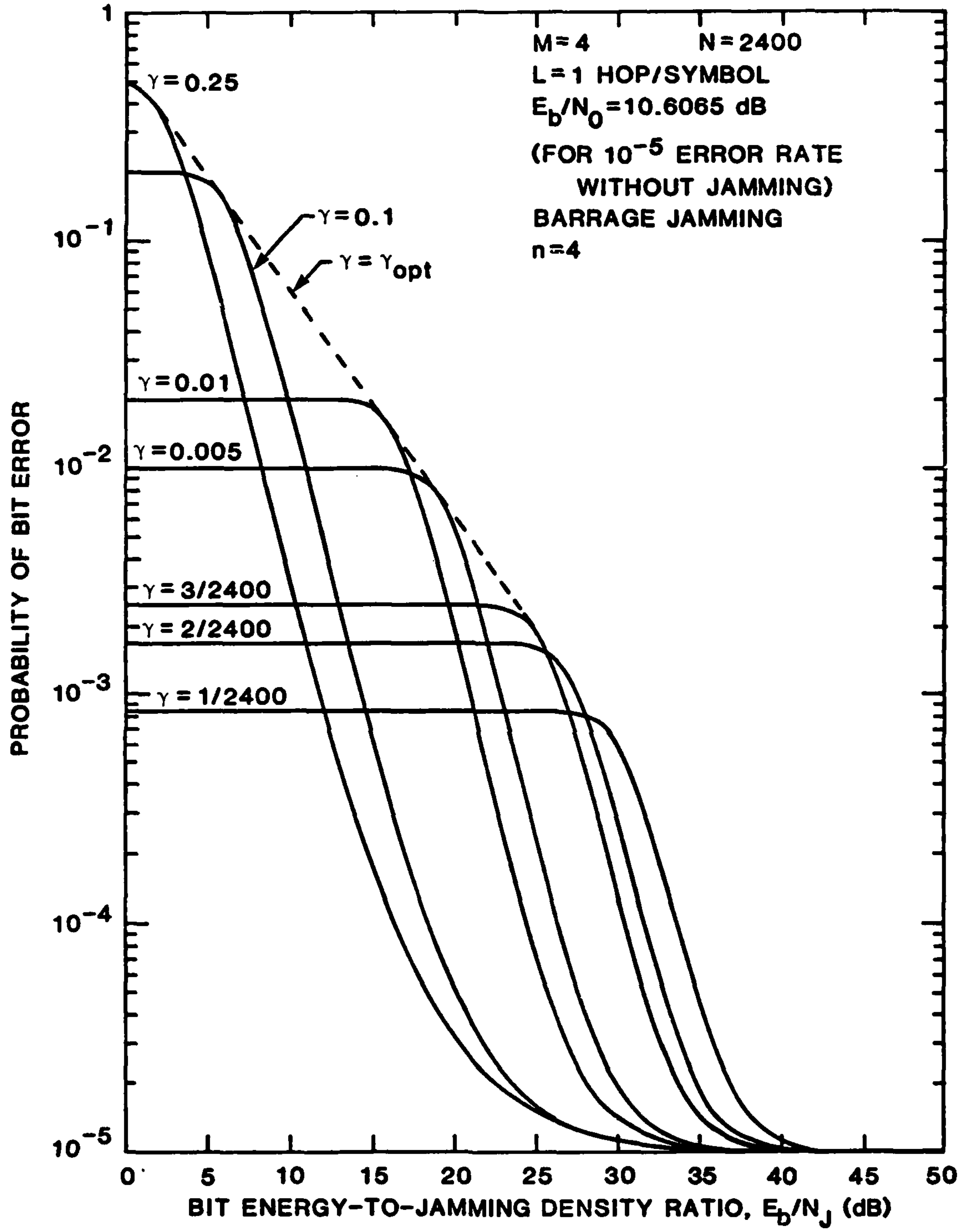


FIGURE 8-35 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n=4) JAMMING AGAINST MFSK/FH FOR M=4 WITH L=1 HOP/SYMBOL, N=2400 HOPPING SLOTS, AND E_b/N_0=10.6065 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

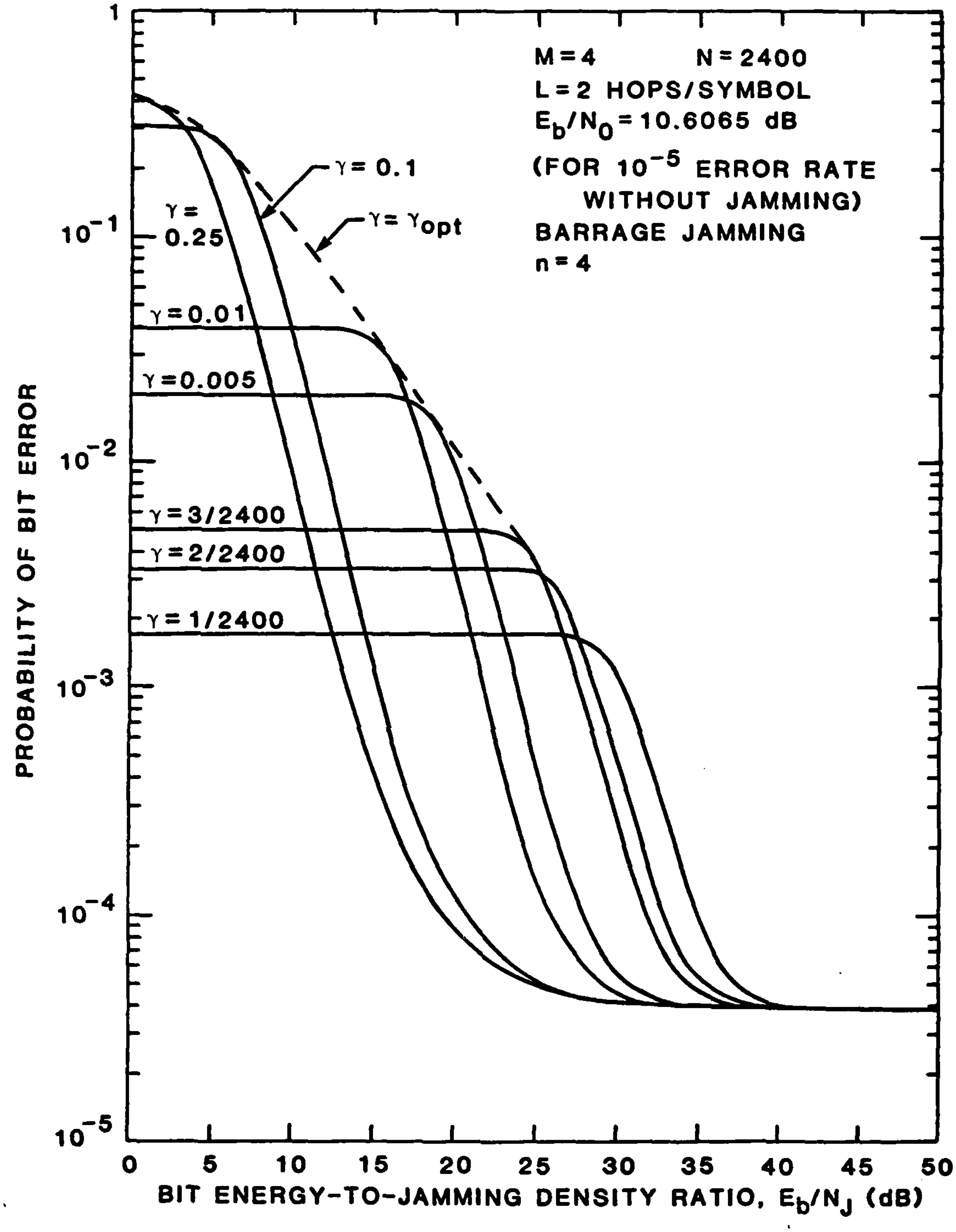


FIGURE 8-36 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n = 4) JAMMING AGAINST MFSK/FH FOR M = 4 WITH L = 2 HOPS/SYMBOL, N = 2400 HOPPING SLOTS, AND E_b/N_0 = 10.6065 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

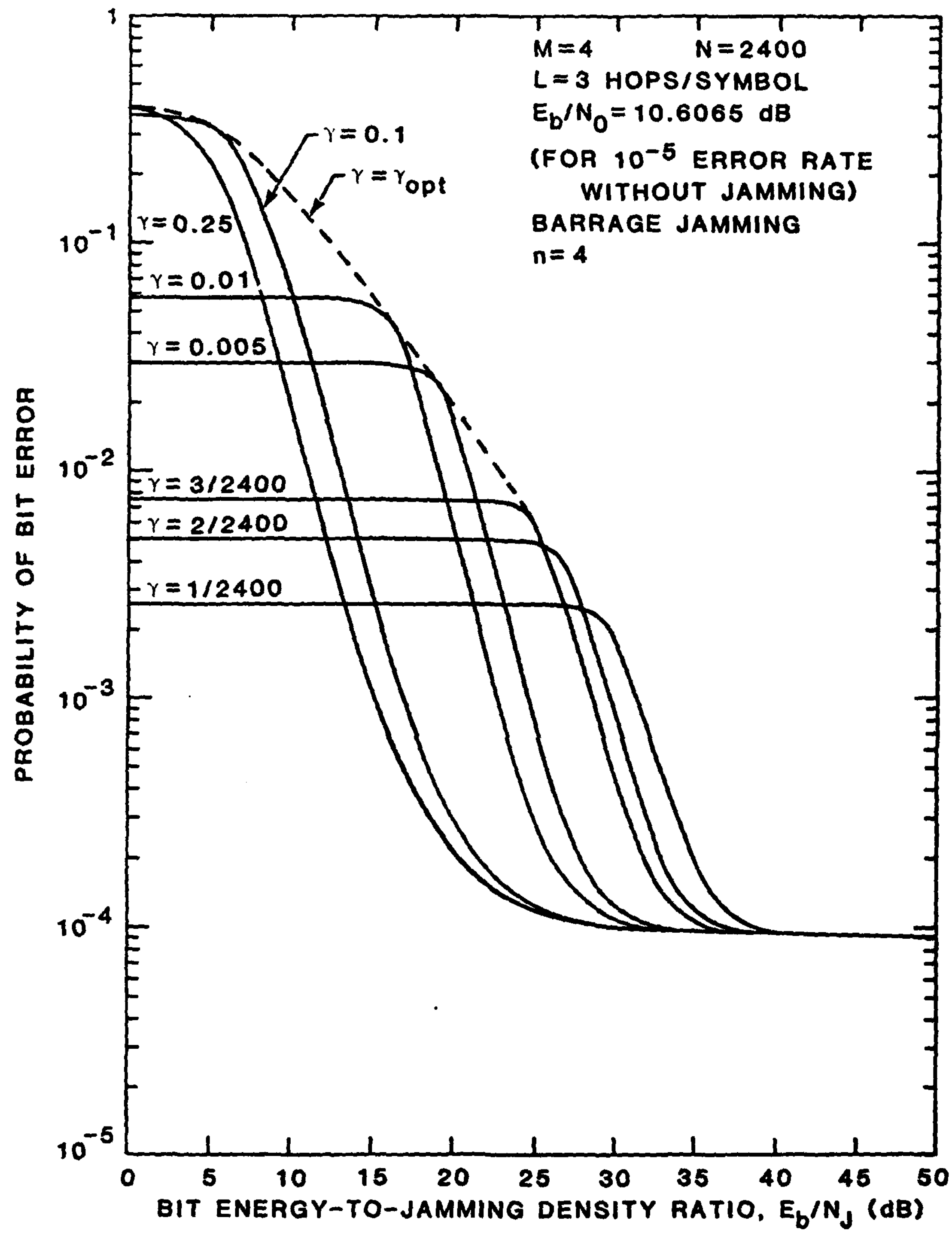


FIGURE 8-37 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n=4) JAMMING AGAINST MFSK/FH FOR M=4 WITH L=3 HOPS/SYMBOL, N=2400 HOPPING SLOTS, AND E_b/N_0=10.6065 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

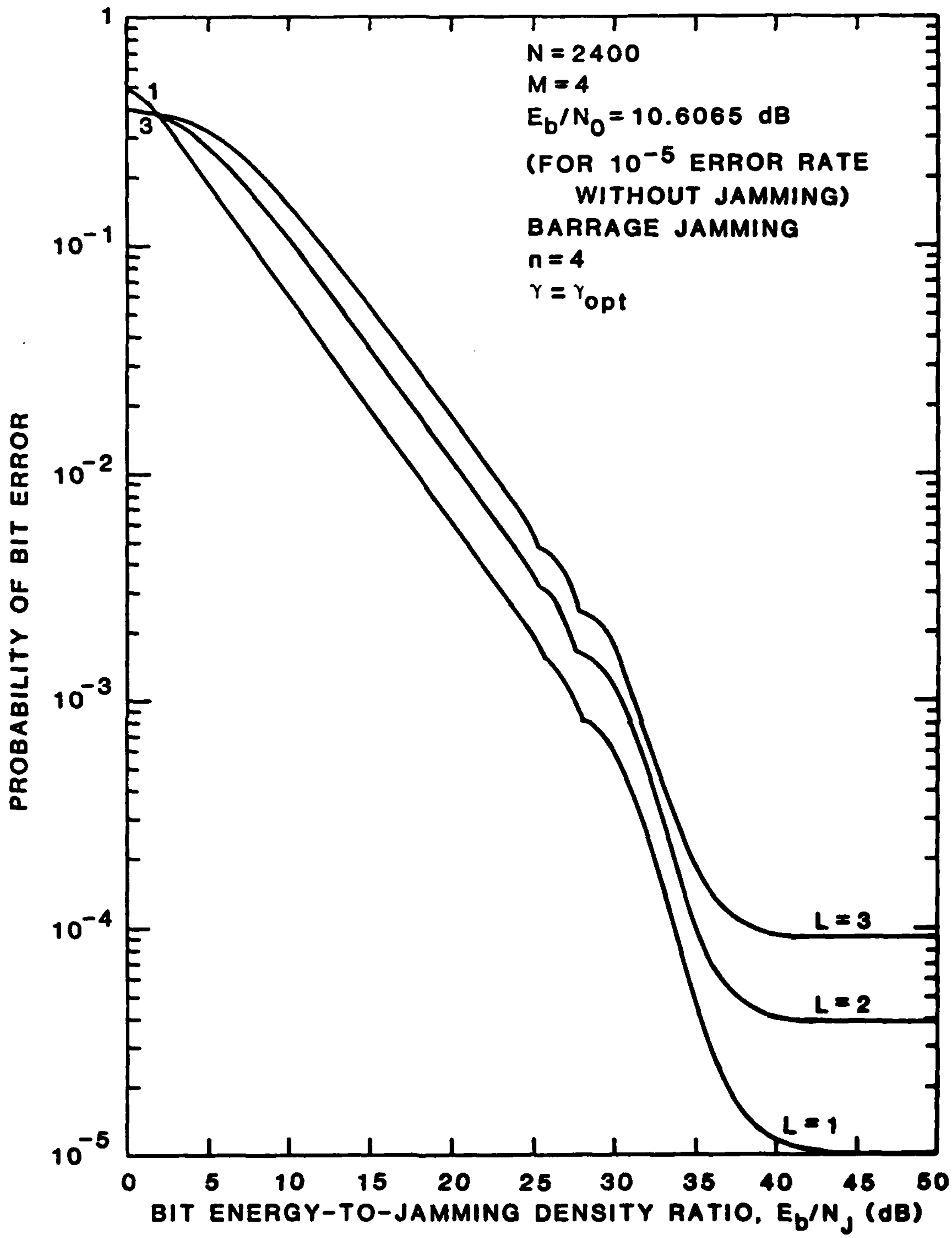


FIGURE 8-38 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH NUMBER OF HOPS PER SYMBOL AS A PARAMETER FOR BARRAGE (n= 4) JAMMING AGAINST MFSK/FH FOR M= 4 WITH OPTIMUM JAMMING FRACTION, N= 2400 HOPPING SLOTS, AND E_b/N_0=10.6065 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

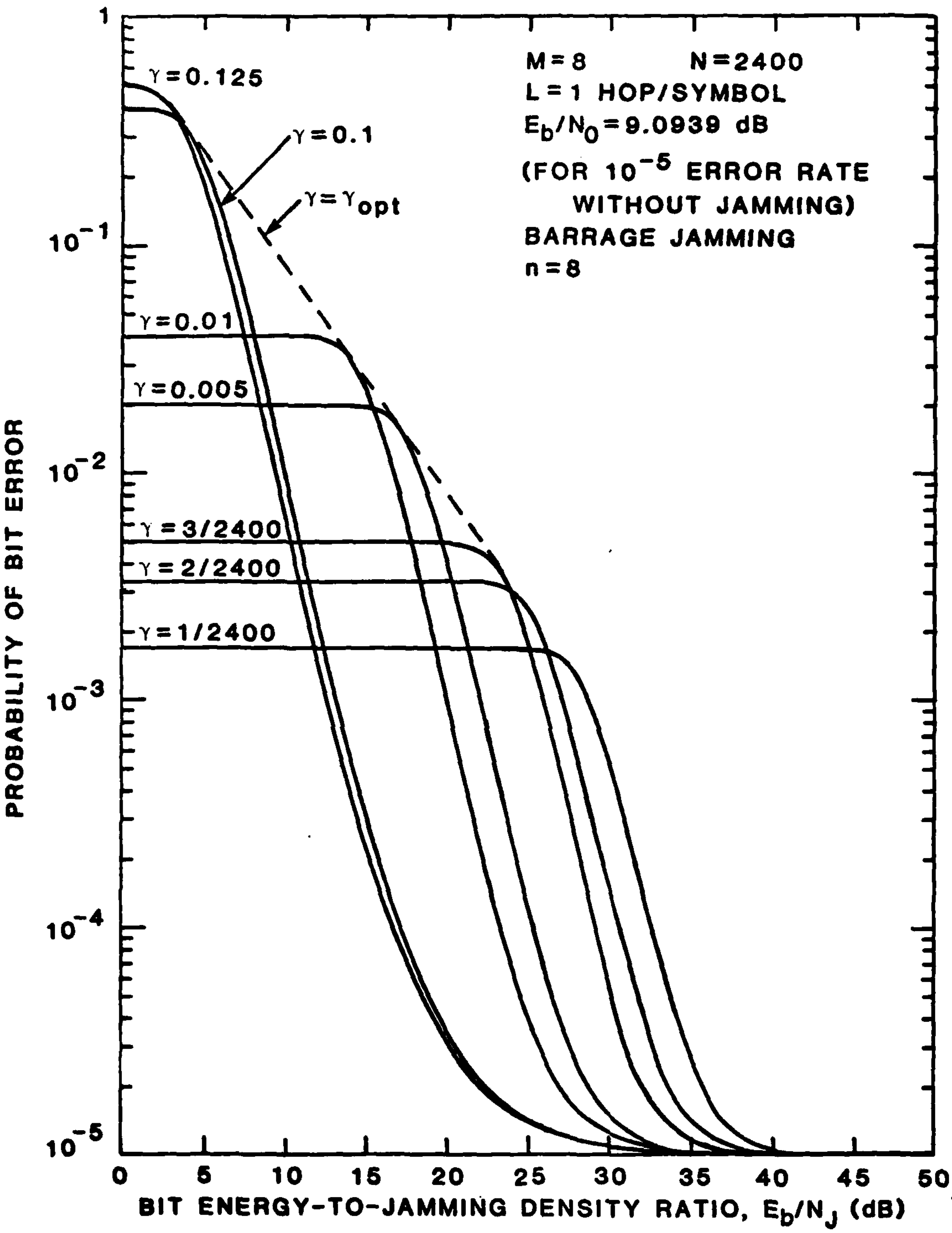


FIGURE 8-39 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n=8) JAMMING AGAINST MFSK/FH FOR M=8 WITH L=1 HOP/SYMBOL, N=2400 HOPPING SLOTS, AND E_b/N_0=9.0939 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

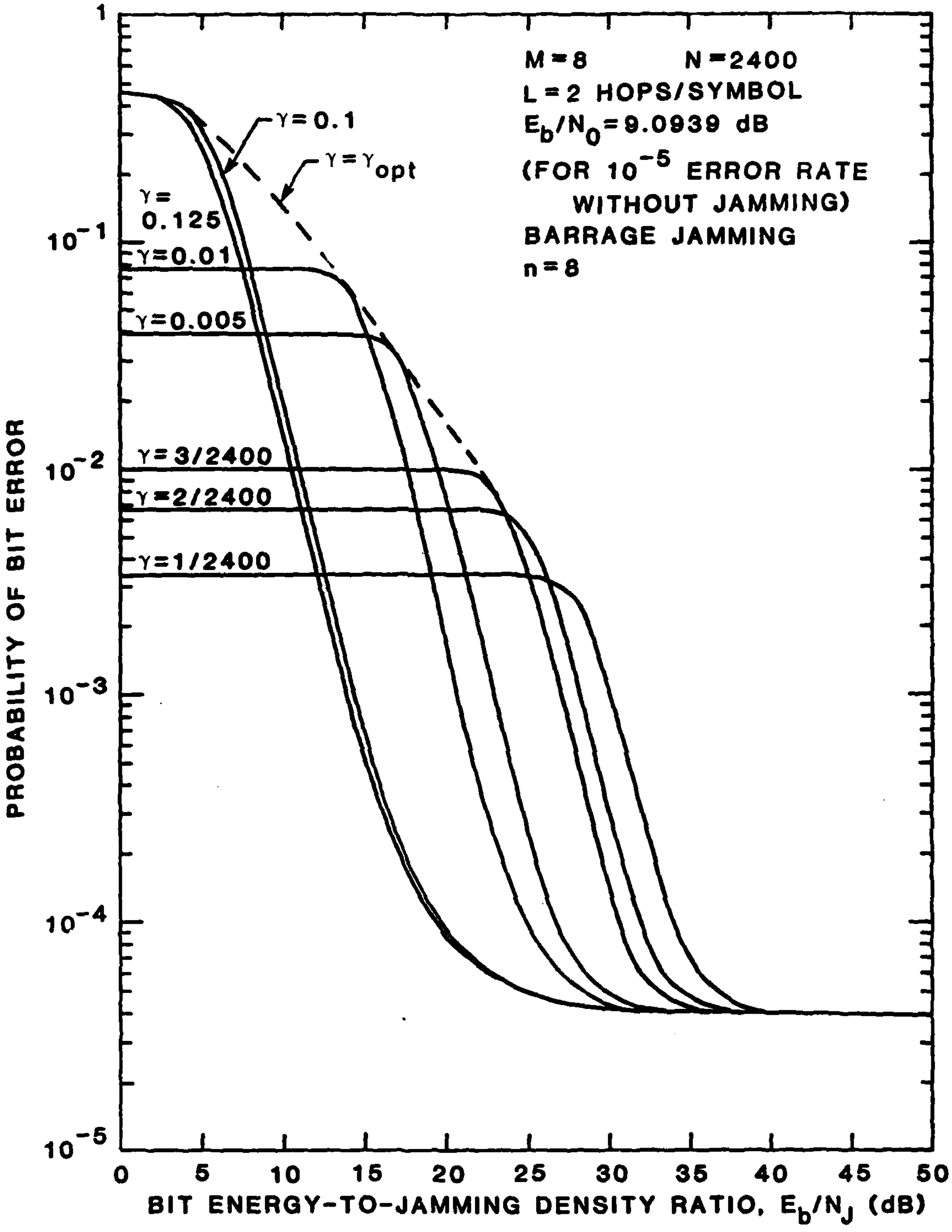


FIGURE 8-40 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH JAMMING FRACTION γ AS A PARAMETER FOR BARRAGE (n=8) JAMMING AGAINST MFSK/FH FOR M=8 WITH L=2 HOPS/SYMBOL, N=2400 HOPPING SLOTS, AND E_b/N_0=9.0939 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

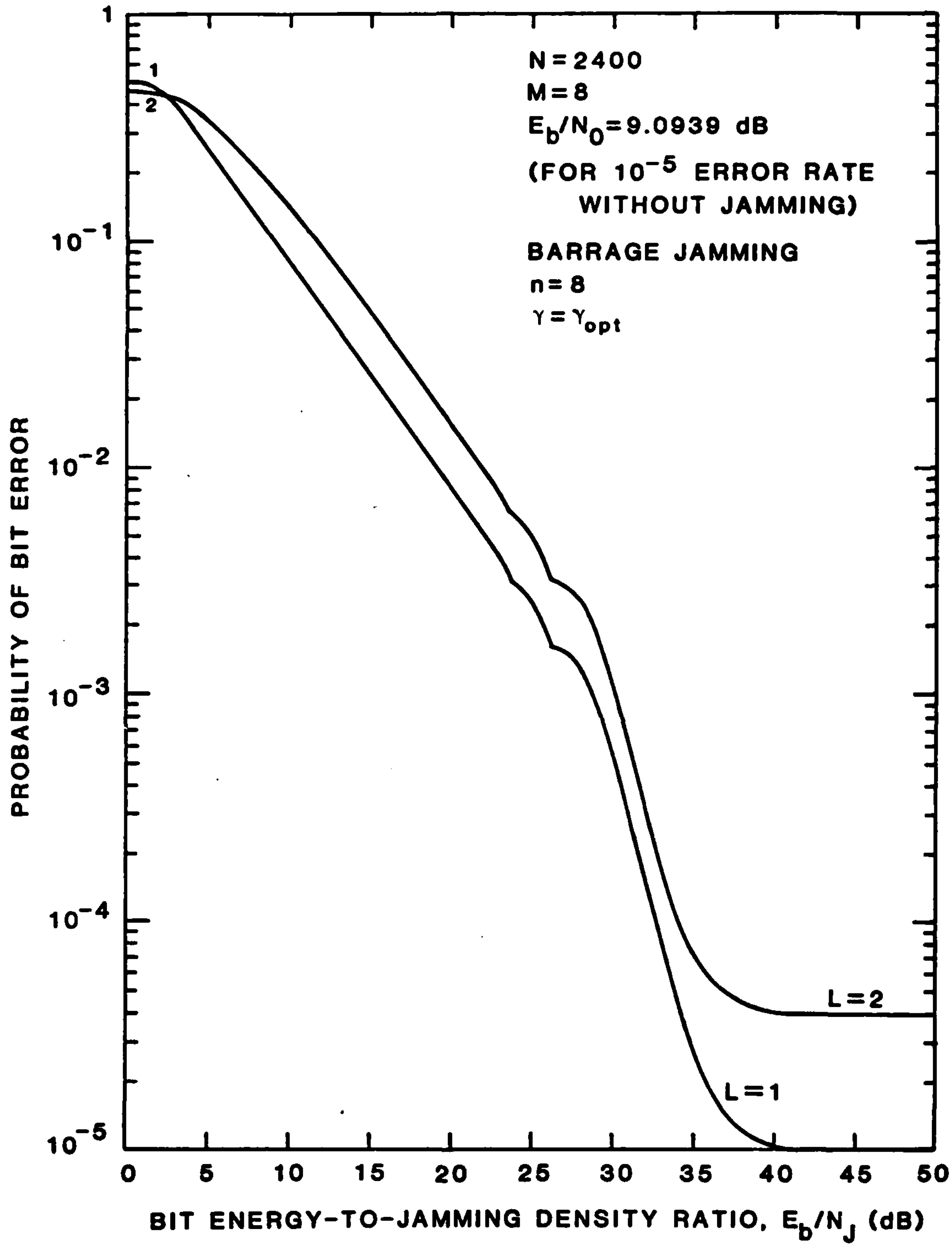


FIGURE 8-41 BIT ERROR PROBABILITY VS. BIT ENERGY-TO-JAMMING DENSITY RATIO WITH NUMBER OF HOPS PER SYMBOL AS A PARAMETER FOR BARRAGE (n=8) JAMMING AGAINST MFSK/FH FOR M=8 WITH OPTIMUM JAMMING FRACTION, N=2400 HOPPING SLOTS, AND E_b/N_0=9.0939 dB (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

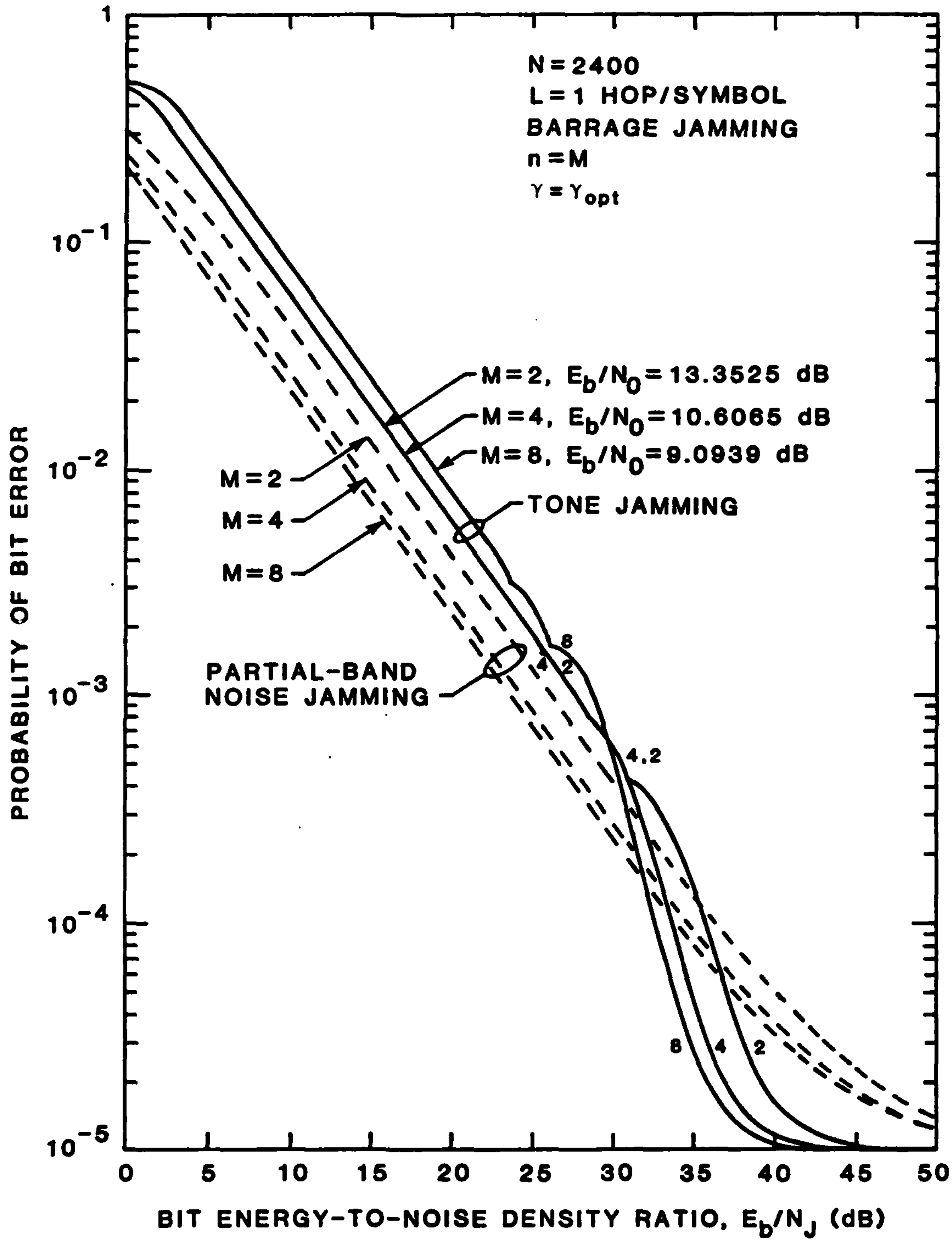


FIGURE 8-42 COMPARISON OF OPTIMUM TONE JAMMING (BARRAGE, n=M) AND OPTIMUM PARTIAL-BAND NOISE JAMMING AGAINST MFSK/FH SQUARE-LAW LINEAR COMBINING RECEIVER WHEN L=1 HOP/SYMBOL AND N=2400 HOPPING SLOTS

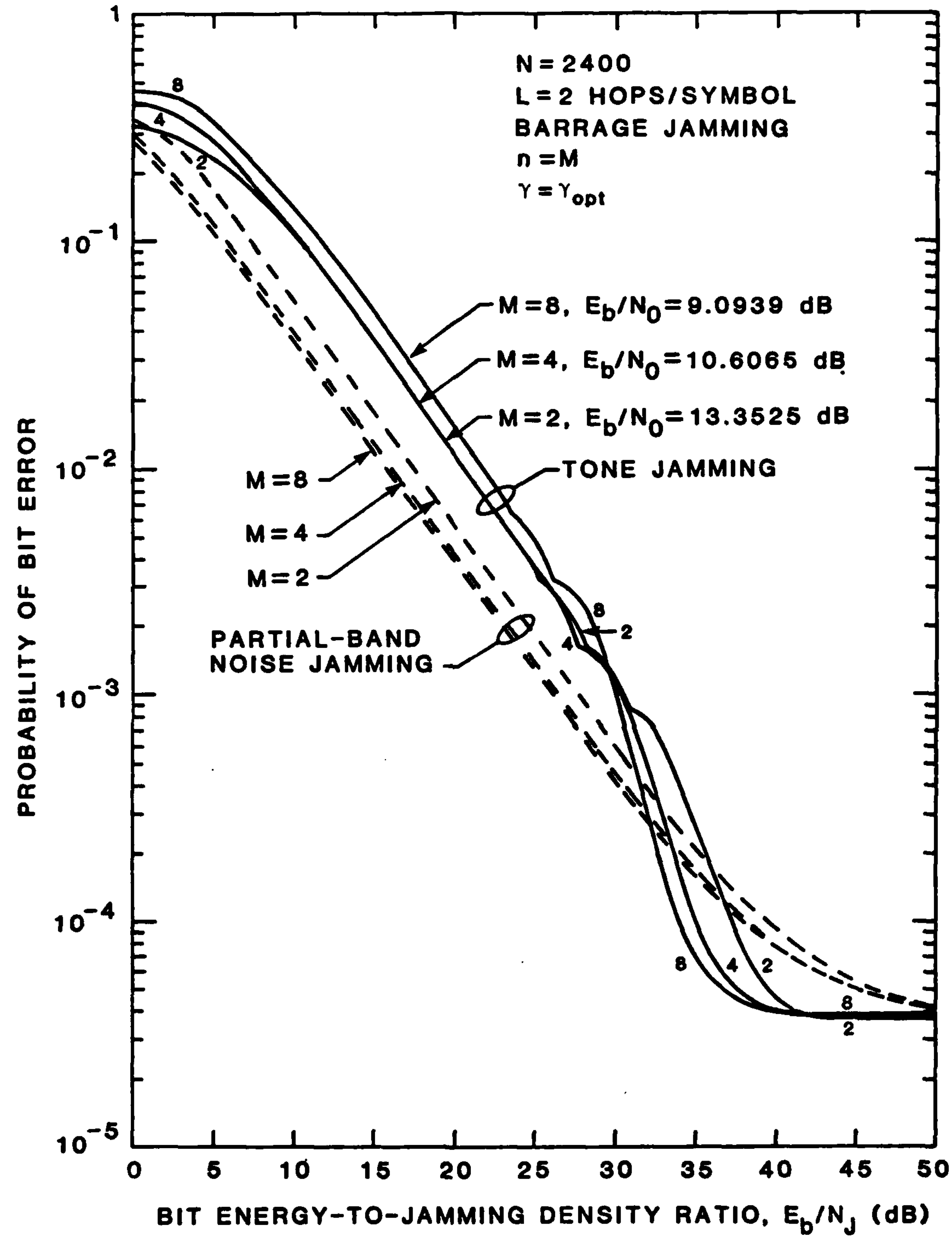


FIGURE 8-43 COMPARISON OF OPTIMUM TONE JAMMING (BARRAGE, n = M) AND OPTIMUM PARTIAL-BAND NOISE JAMMING AGAINST MFSK/FH SQUARE-LAW LINEAR COMBINING RECEIVER WHEN L = 2 HOPS/SYMBOL AND N = 2400 HOPPING SLOTS

band noise jamming. At high E_b/N_J (weak jamming), the partial-band noise jamming appears to be more effective; however this results from a slightly different model used in optimizing the jamming fraction for partial-band noise jamming. In Section 2, we did not impose any lower limit on γ when searching for γ_0. This is equivalent to allowing the number of frequency cells to approach infinity. By taking this approach, we made our analysis of partial-band noise jamming independent of the number of frequency cells, N, in the spread spectrum bandwidth. When we considered tone jamming in this chapter, we were forced by the discrete nature of the tones to recognize the lower limit on γ when optimizing the jamming strategy. If a similar constraint, i.e. $\gamma \geq MB/W = M/N$, were to be imposed on the optimization of the partial-band noise jamming, then these curves would also exhibit a rapid drop-off with increasing values of E_b/N_J and the apparent superiority of partial-band noise jamming in the region of high E_b/N_0 will likely disappear.

One other striking observation is apparent in Figures 8-42 and 8-43. Regardless of the number of hops/symbol, for partial-band noise jamming the communicator gains performance against optimum jamming by increasing M. However just the opposite effect is noted under optimum barrage tone jamming with spacing n = M: as M increases from 4 to 8, the communicator's performance degrades. This same phenomenon is seen in the work of Levitt [13] in the absence of thermal noise.

To understand the mechanism of this behavior under tone jamming, consider the signal-to-jamming power ratios for a fixed value of E_b/N_J and varying M. The energy per bit is

$$E_b = \frac{LS\tau}{K} = \frac{LS}{KB} \tag{8-53}$$

and the jamming density is

$$N_J = \frac{J}{W} = \frac{qJ_0}{W} . \tag{8-54}$$

Therefore,

$$\frac{E_b}{N_J} = \frac{LSW}{KqJ_0} . \tag{8-55}$$

If we let subscript 4 denote quantities associated with the system with $M = 4$ and subscript 8 denote quantities associated with $M = 8$, we have

$$\frac{L_4 S_4 W_4}{2\, q_4\, J_{04}} = \frac{L_8 S_8 W_8}{3\, q_8\, J_{08}} = \frac{E_b}{N_J} = \text{constant} \tag{8-56}$$

where we have used the relation $M = 2^K$. The curves in this chapter have been computed on the basis of $W_4 = W_8$ and $L_4 = L_8$; therefore we may simplify (8-56) by dividing out the W and L factors, yielding

$$\frac{S_4}{2q_4\, J_{04}} = \frac{S_8}{3q_8\, J_{08}} = \text{constant.} \tag{8-57}$$

By rearranging terms in (8-57), we obtain the ratio

$$\frac{(S_4/J_{04})}{(S_8/J_{08})} = \frac{2q_4}{3q_8} . \tag{8-58}$$

If $S_4/J_{04} > S_8/J_{08}$, we would expect the $M = 4$ system to perform better than the $M = 8$ system, since it would exhibit the better signal-to-noise power ratio. This is illustrated in Figure 8-44.

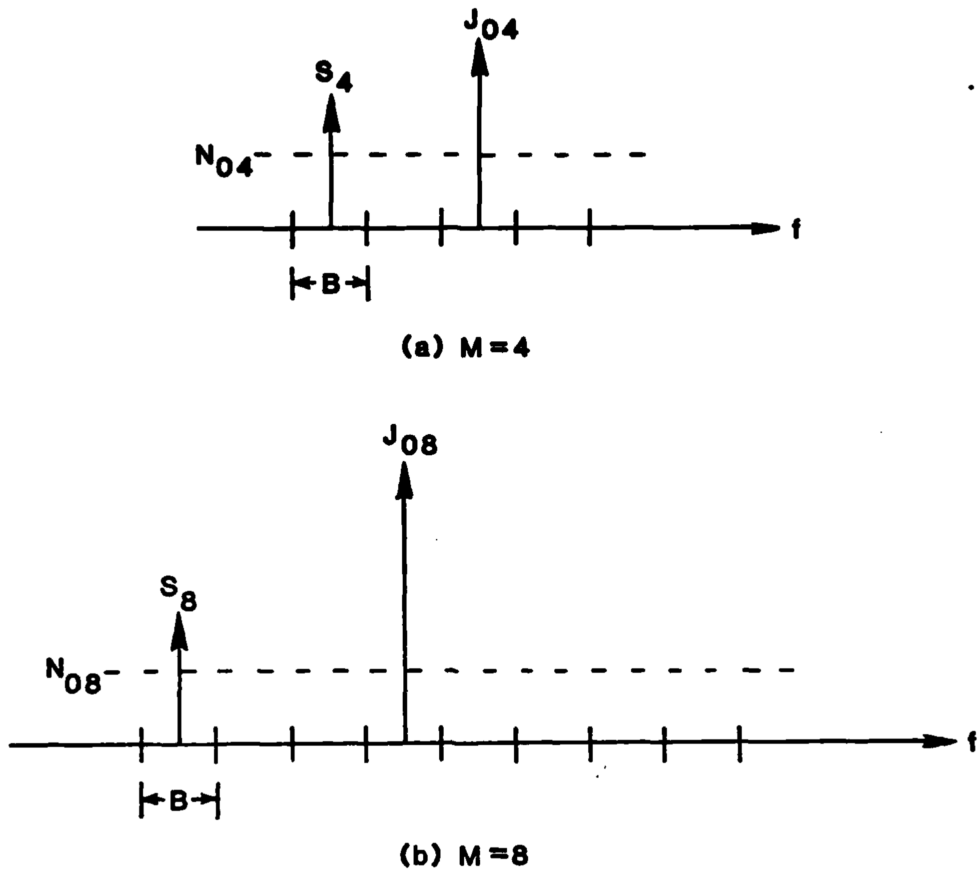


FIGURE 8-44 SIGNAL AND JAMMING POWER LEVELS

From (8-58), we see that

$$\frac{(S_4/J_{04})}{(S_8/J_{08})} > 1 \tag{8-59}$$

if

$$\frac{2q_4}{3q_8} > 1 \tag{8-60a}$$

or

$$\frac{q_4}{q_8} > \frac{3}{2}. \tag{8-60b}$$

Therefore, if the optimum number of jamming tones against the $M = 4$ system is more than 1.5 times the optimum number of jamming tones against the $M = 8$ system, then the $M = 4$ system has a performance advantage.

Thus, we must ask the question, "Can it be that $q_4 > 1.5q_8$ for optimum jamming?" The answer to this question is "yes," as we now show. The optimization of the jamming fraction must take into account not only the conditional error probabilties given a jamming event, but also the probabilities $\Pi_L(\underline{\ell})$ of each jamming event which may occur, given the jamming model and alphabet size M. Let us assume that the optimization process has been performed for $M = 8$ and that the optimum number of jamming tones has been found to be some number, say optimum $q_8 = Q$. This yields a set of one-hop event probabilites $\{\Pi_1(\underline{\ell}_8)\}$ which are used in calculating the total error probability. Although $\{\Pi_1(\underline{\ell}_8)\}$ has 9 elements for barrage tone jamming with $n = 8$ spacing, only 3 distinct types of events need be considered:

$$\Pr\{\text{no channels jammed}|M = 8\} = \Pi_1(0, 0, 0, 0, 0, 0, 0, 0) \tag{8-61a}$$

$$\Pr\{\text{signal channel jammed}|M = 8\} = \Pi_1(1, 0, 0, 0, 0, 0, 0, 0) \tag{8-61b}$$

and

$$\Pr\{\text{any nonsignal channel jammed} | M = 8\} =$$

$$\pi_1(0, 1, 0, 0, 0, 0, 0, 0) + \pi_1(0, 0, 1, 0, 0, 0, 0, 0)$$
$$+ \pi_1(0, 0, 0, 1, 0, 0, 0, 0) + \pi_1(0, 0, 0, 0, 1, 0, 0, 0)$$
$$+ \pi_1(0, 0, 0, 0, 0, 1, 0, 0) + \pi_1(0, 0, 0, 0, 0, 0, 1, 0)$$
$$+ \pi_1(0, 0, 0, 0, 0, 0, 0, 1). \tag{8-61c}$$

It would be reasonable to expect that the optimum jamming fraction for M = 4 would maintain event probabilities close to those for M = 8. To accomplish this, the value of q_4 must be approximately $2.3q_8$, as we show below.

For M = 8 and jamming tones spaced n = 8 filter bandwidths, each jamming tone is capable of corrupting the 8-ary symbol in 8 of its possible hopping positions, as shown in Figure 8-45. Thus we might say that the "span of influence" (to coin a term) of one jamming tone against an 8-ary symbol is 8 hopping positions. As shown in Figure 8-46, the span of influence of a jamming tone against a 4-ary symbol is only 4 hopping positions. The $q_8 = Q$ jamming tones spaced n = 8 filters apart will have non-overlapping spans of influence; hence

$$\Pr\{\text{any nonsignal channel jammed} | M = 8\} = \left(\frac{7}{8}\right) \left(\frac{8Q}{N-7}\right) = \frac{7Q}{N-7} \tag{8-62}$$

and

$$\Pr\{\text{signal channel jammed} | M = 8\} = \left(\frac{1}{8}\right) \left(\frac{8Q}{N-7}\right) = \frac{Q}{N-7} \tag{8-63}$$

for the 8-ary system since each tone influences 8 hopping positions, seven of which place the jamming tone in a non-signal filter. For an M = 4 system, the Q tones would produce only

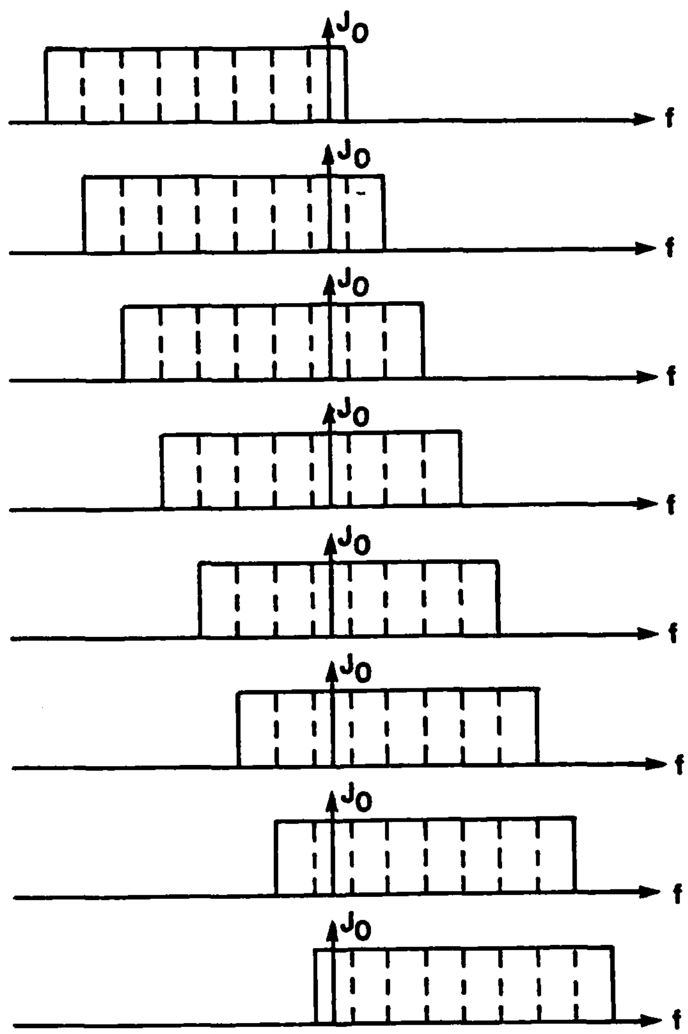


FIGURE 8-45 SPAN OF INFLUENCE OF A JAMMING TONE AGAINST AN 8-ARY SYMBOL

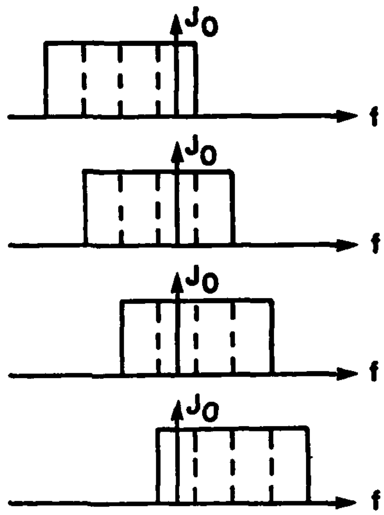


FIGURE 8-46 SPAN OF INFLUENCE OF A JAMMING TONE AGAINST A 4-ARY SYMBOL

$$\Pr\{\text{any nonsignal channel jammed}|M = 4\} = \left(\frac{3}{4}\right)\left(\frac{4Q}{N-3}\right) = \frac{3Q}{N-3} \qquad (8\text{-}64)$$

and

$$\Pr\{\text{signal channel jammed}|M = 4\} = \left(\frac{1}{4}\right)\left(\frac{4Q}{N-3}\right) = \frac{Q}{N-3}\,. \qquad (8\text{-}65)$$

We see from (8-63) and (8-65) that if $N \gg M$, the probabilities that the signal channel is jammed are approximately the same for $M = 4$ and $M = 8$; however this gives a negligible contribution to jamming effectiveness because it will likely reinforce the correct decision. The controlling factors are the probabilities of jamming a non-signal channel. Comparing (8-62) and (8-64), we see that the probability is significantly higher for $M = 8$ with equal numbers of jamming tones. But if Q in (8-64) were replaced by $Q' = 7Q/3 \approx 2.3Q$, then we would have $\Pr\{\text{any nonsignal channel jammed}|M = 4\} \approx \Pr\{\text{any nonsignal channel jammed}|M = 8\}$ for $N \gg M$.

Thus, we see the mechanism by which the communications performance of the $M = 8$ system is degraded relative to the $M = 4$ system. When $M = 8$, the jammer can maintain the probability of influencing the decision with the use of approximately half as many jamming tones as are needed against the $M = 4$ system, thus permitting an increase of the power per tone by approximately a factor of 2. This additional power per jamming tone against the $M = 8$ system permits the jammer to do more damage against the $M = 8$ system than against the $M = 4$ system.

This behavior of MFSK/FH systems under barrage tone jamming with $n = M$ illustrates vividly the intelligence requirements of the jammer. The jammer must know the alphabet size, M, in order to optimize the jamming strategy to maximize the effectiveness of the countermeasure.

APPENDIX 1A

COMMENTS OF GENERAL APPLICABILITY TO COMPUTER PROGRAM LISTINGS

A number of computer program listings are given in Appendices 2G, 2H, 2I, 4E, 4F, 4G, 4H, 4I, 5A, 5C, 5E, 8C, 8F, 8G, 8H, and 8I. These listings were produced by manually editing the listings produced by the DEC FORTRAN-77 V4.0-1 compiler and then printing them on a letter-quality printer. The editing consisted of deletion of compiler-produced storage maps and removal of file names from the remaining output. The pagination was then adjusted to meet the page-size requirements of this report, and the listing page numbers were adjusted to be continuous. Other than this editing for purposes of mechanical format, no changes were made to the programs themselves; the listings accurately reflect the results of an error-free compilation.

Many of the programs make use of a few extensions to the standard FORTRAN-77 provided by the DEC compiler. These are decribed below for the benefit of those who may desire to run these programs on other systems.

In a FORMAT statement, the angle brackets enclosing an integer expression, e.g. <N>, permit the use of integer variables or expressions wherever standard FORTRAN-77 permits an integer constant in the FORMAT. We have used this language extension in output formats to provide a variable number of columns and column headings, as a function of the number of cases run.

Also in FORMAT statements, we have made frequent use of the format item $. This format item suppresses a carriage return at the end of the record. We use it when writing prompts for input at the start of a run.

Some programs make use of the VIRTUAL statement, which declares an array to reside outside the program's normal address space (which is limited to 64K bytes). This allows use of large arrays, up to the available physical memory. On systems without the address space restriction, e.g. IBM 370 systems, the VIRTUAL statement may be changed to a DIMENSION statement.

Several calls to system-supplied subroutines are used in a number of the programs. These are summarized and described below, except for mathematical functions where are described with the appendix which uses them.

ERRSET	Set error action; used in our programs to enable floating underflow messages or (in one case) to make floating overflows immediately fatal
GETADR	Return the address of a variable; used in doing non-standard I/O operations
GETLUN	Get characteristics of a device associated with a logical unit
SECNDS	Return time since midnight in seconds; used in some of our programs to measure execution time
WTQIO	Request system I/O operation and wait for completion; bypasses FORTRAN I/O system to perform operations not available through the standard FORTRAN I/O package

The reference to TI: in some comments in the programs is to the pseudo-device "terminal of issuance," i.e the terminal from which the program is run.

The logical unit assignments used on our system are as follows: 5--the terminal from which the program is run; 6--either a disk file (usually) or the printing terminal (optionally), as defined during task building. Any other logical units are disk files.

APPENDIX 2A

PARTIAL-FRACTION EXPANSION OF (2-64)

We desire a partial-fraction expansion of the characteristic function

$$\Phi(j\nu) = \frac{1}{(1-j2\sigma_T^2\nu)^{\ell}\,(1-j2\sigma_N^2\nu)^{L-\ell}}\,. \tag{2A-1}$$

To simplify the notation, we define

$$a_T \triangleq \frac{1}{2\sigma_T^2}\,, \tag{2A-2a}$$

$$a_N \triangleq \frac{1}{2\sigma_N^2}\,, \tag{2A-2b}$$

and

$$s \triangleq j\nu\,. \tag{2A-2c}$$

With these substitutions, we can write (2A-1) in the standard form

$$\Phi(s) = (-1)^L\, a_T^{\ell}\, a_N^{L-\ell}\, \frac{1}{(s-a_T)^{\ell}\,(s-a_N)^{L-\ell}} \tag{2A-3}$$

which admits to a partial-fraction expansion

$$\Phi(s) = (-1)^L\, a_T^{\ell}\, a_N^{L-\ell}\left[\sum_{r=1}^{\ell} \frac{A_r}{(s-a_T)^r} + \sum_{r=1}^{L-\ell} \frac{B_r}{(s-a_N)^r}\right] \tag{2A-4}$$

provided neither $\ell=0$ nor $L-\ell=0$. We may accommodate these exceptional cases (which are mutually exclusive since $L > 0$) by defining $A_0 = B_0 = 0$ and starting the summations at $r=0$ so as to have an explicit 0 for the empty summation.

The coefficients A_r and B_r in (2A-4) may be determined by the formulas [15, pp. 46-47]

$$A_r = \frac{1}{(\ell-r)!} \left\{ \frac{d^{\ell-r}}{ds^{\ell-r}} \left[\frac{1}{(s - a_N)^{L-\ell}} \right] \right\} \Bigg|_{s=a_T} \tag{2A-5}$$

and

$$B_r = \frac{1}{(L-\ell-r)!} \left\{ \frac{d^{L-\ell-r}}{ds^{L-\ell-r}} \left[\frac{1}{(s - a_T)^{\ell}} \right] \right\} \Bigg|_{s=a_N} . \tag{2A-6}$$

The derivatives in (2A-5) and (2A-6) may be evaluated using the formula [16, p. 86]

$$\frac{d^N}{ds^N} s^{-K} = (-1)^N (K)_N s^{-K-N} \tag{2A-7}$$

where the Pochhammer symbol is defined as [4, eq. 6.1.22]

$$(K)_0 = 1 \tag{2A-8a}$$

$$(K)_N = \Gamma(K + N)/\Gamma(K). \tag{2A-8b}$$

Using (2A-7) to evaluate (2A-5) and (2A-6), and substituting the parameters defined in (2A-2), we find that the coefficients of the partial-fraction expansion are given by

$$A_r = \frac{(-1)^{\ell-r}(L-\ell)_{\ell-r}}{(\ell-r)!} \left(\frac{1}{1-\delta}\right)^{L-r} (2\sigma_T^2)^{L-r} , \quad r = 1, 2,\ldots,\ell \tag{2A-9}$$

and

$$B_r = \frac{(-1)^{L-\ell-r}(\ell)_{L-\ell-r}}{(L-\ell-r)!} \left(\frac{1}{\delta-1}\right)^{L-r} (2\sigma_T^2)^{L-r}, \quad r = 1, 2,\ldots,L-\ell \tag{2A-10}$$

where we define

$$\delta \triangleq \sigma_T^2/\sigma_N^2 \ . \tag{2A-11}$$

When we apply the definitions in (2A-2) to the factors $(s - a_T)^{-r}$ and $(s - a_N)^{-r}$ which occur in (2A-4), we obtain

$$\frac{1}{(s - a_T)^r} = \frac{(-1)^r (2\sigma_T^2)^r}{(1 - j2\sigma_T^2 \nu)^r} \tag{2A-12}$$

and

$$\frac{1}{(s - a_N)^r} = \frac{(-1)^r (2\sigma_N^2)^r}{(1 - j2\sigma_N^2 \nu)^r} \ . \tag{2A-13}$$

Using (2A-12) and (2A-13) in (2A-4), we obtain

$$\phi(j\nu) = (-1)^{\ell} \left(\frac{1}{2\sigma_T^2}\right)^{\ell} \left(\frac{1}{2\sigma_N^2}\right)^{L-\ell} \left[\sum_{r=0}^{\ell} \frac{(-1)^r (2\sigma_T^2)^r \, A_r}{(1-j2\sigma_T^2\nu)^r} + \sum_{r=0}^{L-\ell} \frac{(-1)^r (2\sigma_N^2)^r \, B_r}{(1-j2\sigma_N^2\nu)^r}\right] \tag{2A-14}$$

where

$$A_0 = B_0 = 0, \tag{2A-15a}$$

$$A_r = \frac{(-1)^{\ell-r} (L-\ell)_{\ell-r}}{(\ell-r)!} \left(\frac{1}{1-\delta}\right)^{L-r} \left(2\sigma_T^2\right)^{L-r}, \ r = 1, 2, \ldots, \ell, \tag{2A-15b}$$

and

$$B_r = \frac{(-1)^{L-\ell-r} (\ell)_{L-\ell-r}}{(L-\ell-r)!} \left(\frac{1}{\delta-1}\right)^{L-r} \left(2\sigma_T^2\right)^{L-r}, \ r = 1, 2, \ldots, L-\ell. \tag{2A-15c}$$

APPENDIX 2B

PARTIAL-FRACTION EXPANSION OF (2-78)

We desire a partial-band expansion of the function

$$\psi_{m,n}(j\nu) = \frac{1}{(1-j2\sigma_T^2\nu)^{\ell+m}(1-j2\sigma_N^2\nu)^{L-\ell+n}} . \tag{2B-1}$$

To simplify the notation, we define

$$a_T \triangleq \frac{1}{2\sigma_T^2} , \tag{2B-2a}$$

$$a_N \triangleq \frac{1}{2\sigma_N^2} , \tag{2B-2b}$$

and

$$s \triangleq j\nu . \tag{2B-2c}$$

With these substitutions, we can write (2B-1) in the standard form

$$\psi_{m,n}(s) = (-1)^{L+m+n} a_T^{\ell+m} a_N^{L-\ell+n} \frac{1}{(s-a_T)^{\ell+m}(s-a_N)^{L-\ell+n}} \tag{2B-3}$$

which admits to a partial-fraction expansion

$$\psi_{m,n}(s) = (-1)^{L+m+n} a_T^{\ell+m} a_N^{L-\ell+n} \left[\sum_{r=1}^{\ell+m} \frac{C_r}{(s-a_T)^r} + \sum_{r=1}^{L-\ell+n} \frac{D_r}{(s-a_N)^r} \right] \tag{2B-4}$$

provide neither $\ell+m=0$ nor $L-\ell+n=0$. We may accomodate these exceptional cases (which are mutually exclusive since $L>0$, $\ell\geq 0$, $m\geq 0$, $n\geq 0$) by defining

$C_0=D_0=0$ and starting the summations at r=0 so as to have an explicit 0 for the empty summation.

The coefficients C_r and D_r in (2B-4) may be determined by the formulas [15, pp. 46-47]

$$C_r = \frac{1}{(\ell+m-r)!} \left\{ \frac{d^{\ell+m-r}}{ds^{\ell+m-r}} \left[\frac{1}{(s-a_N)^{L-\ell+n}} \right] \right\} \Bigg|_{s=a_T} \tag{2B-5}$$

and

$$D_r = \frac{1}{(L-\ell+n-r)!} \left\{ \frac{d^{L-\ell+n-r}}{ds^{L-\ell+n-r}} \left[\frac{1}{(s-a_T)^{\ell+m}} \right] \right\} \Bigg|_{s=a_N} \tag{2B-6}$$

The derivatives in (2B-5) and (2B-6) may be evaluated using the formula [16, p. 86]

$$\frac{d^N}{ds^N} s^{-K} = (-1)^N (K)_N s^{-K-N} \tag{2B-7}$$

where the Pochhammer symbol is defined as [4, eq. 6.1.22]

$$(K)_0 = 1 \tag{2B-8a}$$

$$(K)_N = \Gamma(K+N)/\Gamma(K). \tag{2B-8b}$$

Using (2B-7) to evaluate (2B-5) and (2B-6), and substituting the parameters defined in (2B-2), we find that the coefficients of the partial-fraction expansion are given by

$$C_r = \frac{(-1)^{L-\ell+n}(L-\ell+n)_{\ell+m-r}}{(\ell+m-r)!} \left(\frac{2\sigma_T^2}{\delta-1}\right)^{L+m+n-r} \tag{2B-9}$$

and

$$D_r = \frac{(-1)^{L-\ell+n-r}(\ell+m)_{L-\ell+n-r}}{(L-\ell+n-r)!}\left(\frac{2\sigma_T^2}{\delta-1}\right)^{L+m+n-r} \tag{2B-10}$$

where we define

$$\delta \triangleq \sigma_T^2/\sigma_N^2. \tag{2B-11}$$

When we apply the definitions in (2B-2) to the factors $(s-a_T)^{-r}$ and $(s-a_N)^{-r}$ which occur in (2B-4), we obtain

$$\frac{1}{(s-a_T)^r} = \frac{(-1)^r(2\sigma_T^2)^r}{(1-j2\sigma_T^2\nu)^r} \tag{2B-12}$$

and

$$\frac{1}{(s-a_N)^r} = \frac{(-1)^r(2\sigma_N^2)^r}{(1-j2\sigma_N^2\nu)^r}. \tag{2B-13}$$

Using (2B-12) and (2B-13) in (2B-4), we obtain

$$\psi_{m,n}(j\nu) = (-1)^{L+m+n}\left(\frac{1}{2\sigma_T^2}\right)^{\ell+m}\left(\frac{1}{2\sigma_N^2}\right)^{L-\ell+n}\left[\sum_{r=0}^{\ell+m} C_r \frac{(-1)^r(2\sigma_T^2)^r}{(1-j2\sigma_T^2\nu)^r} + \sum_{r=0}^{L-\ell+n} D_r \frac{(-1)^r(2\sigma_N^2)^r}{(1-j2\sigma_N^2\nu)^r}\right]. \tag{2B-14}$$

APPENDIX 2C

AN EXPRESSION FOR M-ARY FSK/FH SYMBOL ERROR PROBABILITY FOR THE SQUARE-LAW LINEAR COMBINING RECEIVER WHICH DOES NOT CONTAIN AN INTEGRAL TO BE EVALUATED

In the main text we arrived at an expression for the MFSK/FH symbol error probability in which one integral remained to be evaluated, namely (2-87). In principle, this integral is readily evaluated; but in detail process is very tedious and the result is so complicated that it is of little computational utility. Notwithstanding this, we pursue this final integration to its conclusion for the benefit of those readers who prefer not leave an integral in the final result.

We begin with the form from the main text, which we repeat here for ready reference:

$$P_s(e) = 1 - \sum_{\ell=0}^{L} (-1)^{\ell M} \binom{L}{\ell} \gamma^{\ell}(1-\gamma)^{L-\ell} e^{-\ell\rho_T} e^{-(L-\ell)\rho_N} \sum_{m=0}^{\infty} \frac{(\ell\rho_T)^m}{m!} \sum_{n=0}^{\infty} \frac{[(L-\ell)\rho_N]^n}{n!}$$

$$\cdot \int_0^{\infty} \left\{ \sum_{r=0}^{\ell+m} (1-\delta_{r,0}) \frac{(-1)^{m-r}(L-\ell+n)_{\ell+m-r}}{(\ell+m-r)!} \delta^{L-\ell+n} \left(\frac{1}{\delta-1}\right)^{L+m+n-r} \frac{1}{\Gamma(r)} x^{r-1} e^{-x} \right.$$

$$\left. + \sum_{r=0}^{L-\ell+n} (1-\delta_{r,0}) \frac{(-1)^m(\ell+m)_{L-\ell+n-r}}{(L-\ell+n-r)!} \delta^{L-\ell+n} \left(\frac{1}{\delta-1}\right)^{L+m+n-r} \frac{1}{\Gamma(r)} x^{r-1} e^{-\delta x} \right\}$$

$$\cdot \left\{ \sum_{r=0}^{\ell} (1-\delta_{r,0}) \frac{(-1)^r(L-\ell)_{\ell-r}}{(\ell-r)!} \delta^{L-\ell} \left(\frac{1}{\delta-1}\right)^{L-r} P(r,x) \right.$$

$$+ \sum_{r=0}^{L-\ell} (1-\delta_{r,0}) \frac{(\ell)_{L-\ell-r}}{(L-\ell-r)!} \left(\frac{1}{\delta-1}\right)^{L-r} \delta^{L-\ell-r} \; P(r,\delta x)\Bigg\}^{M-1} dx. \qquad (2C\text{-}1)$$

We will begin by making some observations which will be used to simplify notation in the following steps. First, we note that the form $(M)_N/N!$ can be written as a binomial coefficient. To show this, we have the progression

$$\frac{(M)_N}{N!} = \frac{\Gamma(M+N)}{\Gamma(M)N!}$$

$$= \frac{(M+N-1)!}{(M-1)!N!}$$

$$= \frac{(M-1+N)!}{(M-1)!N!}$$

$$= \binom{M-1}{N}. \qquad (2C\text{-}2)$$

Second, we will be forming powers of finite sums in the process of evaluating the (M-1)-st power which occurs in (2C-1). That is, we will have things of the form

$$X = \left(\sum_{i=0}^{M} x_i\right)^N \qquad (2C\text{-}3)$$

which must be expressed in terms of sums of products of the x_i's. By the multinomial theorem [4, Sec. 24.1.2],

$$X = \sum \binom{N}{n_0,\ n_1, \ldots, n_M} x_0^{n_0}\, x_1^{n_1} \ldots x_M^{n_M} \qquad (2C\text{-}4a)$$

where the summation is taken over all combinations of non-negative indices n_i, $i = 1,2,\ldots,M$, for which

$$\sum_{i=0}^{M} n_i = N \tag{2C-4b}$$

and the multinomial coefficients are given by

$$\binom{N}{n_0, n_1, \ldots, N_M} = \frac{N!}{n_0!\, n_1! \ldots n_M!} = \frac{N!}{\prod_{i=0}^{M} n_i!} \,. \tag{2C-4c}$$

Substitution of (2C-4c) into (2C-4a) yields

$$X = \sum_{\{n_i:(0,M);\Sigma=N\}} \left(N! \prod_{j=0}^{M} \frac{x_j^{n_j}}{n_j!} \right) \tag{2C-5}$$

where we have introduced the shorthand notation* that, formally, if $M \geq 0$ then

$$\sum_{\{n_i:(0,M);\Sigma=N\}} \triangleq \underset{\sum_{i=0}^{M} n_i = N}{\sum_{n_0=0}^{N} \sum_{n_1=0}^{N} \cdots \sum_{n_M=0}^{N}} \tag{2C-6}$$

and if $M<0$ then the left-hand side of (2C-6) is taken to be a null operator. Here the notation $\{n_i:(0,M);\Sigma=N\}$ is defined to mean that the summation is taken over indices n_i, $i=0,1,\ldots,M$, subject to the constraint (2C-4b).

We also will find it useful to express the incomplete gamma function $P(N,x)$ as a finite sum of terms [4, 6.5.13]:

$$P(N,x) = 1-e^{-x} \sum_{i=0}^{N-1} \frac{x^i}{i!} \,. \tag{2C-7}$$

Our goal in the subsequent analysis will be to obtain a form in which each term is of the form $\alpha x^N e^{-\beta x}$ where α and β do not depend upon x so that the result may be integrated term-by-term using the formula [2, eq. 3.381.4]

*We take the position that any reasonable shorthand notation which will reduce the size of the final form is desirable.

$$\int_0^\infty x^{\nu-1} e^{-\mu x} dx = \frac{\Gamma(\nu)}{\mu^\nu} . \tag{2C-8}$$

We begin by using the binomial theorem to expand the (M-1)-st power which occurs in (2C-1). Let Q^{M-1} denote this portion of (2C-1). Then, using (2C-2),

$$Q^{M-1} = \delta^{(M-1)(L-\ell)} \left(\frac{1}{\delta-1}\right)^{L(M-1)} \sum_{k=0}^{M-1} \binom{M-1}{k} Q_1^k \; Q_2^{M-1-k} \tag{2C-9}$$

where

$$Q_1 \triangleq \sum_{r_1=0}^{\ell} (1-\delta_{r_1,0}) \binom{L-\ell-1}{\ell-r_1} (-1)^{r_1} (\delta-1)^{r_1} P(r_1,x) \tag{2C-10a}$$

and

$$Q_2 \triangleq \sum_{r_2=0}^{L-\ell} (1-\delta_{r_2,0}) \binom{\ell-1}{L-\ell-r} \left(\frac{\delta-1}{\delta}\right)^{r_2} P(r_2,\delta x). \tag{2C-10b}$$

We use (2C-5) to express Q_1^k and Q_2^{M-1-k} with the results

$$Q_1^k = \sum_{\{\mu_j:(0,\ell);\Sigma=k\}} k! \prod_{j=0}^{\ell} \frac{(-1)^{j\mu_j}}{\mu_j!} \left[(1-\delta_{j,0}) \binom{L-\ell-1}{\ell-j} (\delta-1)^j\right]^{\mu_j} \left[P(j,x)\right]^{\mu_j} \tag{2C-11a}$$

and

$$Q_2^{M-1-k} = \sum_{\{\nu_j:(0,L-\ell);\Sigma=M-1-k\}} (M-1-k)! \prod_{\lambda=0}^{L-\ell} \frac{1}{\nu_\lambda!} \left[(1-\delta_{\lambda,0})\binom{L-1}{L-\ell-\lambda}\left(\frac{\delta-1}{\delta}\right)^{\lambda}\right]^{\nu_\lambda}$$

$$\cdot \left[P(\lambda,\delta x)\right]^{\nu_\lambda}. \tag{2C-11b}$$

We need a means of expressing the powers of the incomplete gamma function. Using (2C-7) and the binomial theorem, we can write

$$[P(j,x)]^{\mu_j} = \left(1 - e^{-x} \sum_{w=0}^{j-1} \frac{x^w}{w!}\right)^{\mu_j}$$

$$= \sum_{v=0}^{\mu_j} \binom{\mu_j}{v} (-1)^v \left(\sum_{w=0}^{j-1} \frac{x^w}{w!} \right)^v e^{-vx} \qquad (2C\text{-}12)$$

where an empty sum of the form $\sum_{w=0}^{-1}$ is taken to be zero. Now we apply (2C-5) to (2C-12) to obtain

$$[P(j,x)]^{\mu_j} = \sum_{v=0}^{\mu_j} \binom{\mu_j}{v} (-1)^v \, e^{-vx} \sum_{\{\alpha_i:(0,j-1);\Sigma=\mu_j\}} \mu_j! \prod_{w=0}^{j-1} \frac{x^{w\alpha_w}}{\alpha_w (w!)^{\alpha_w}} \qquad (2C\text{-}13)$$

where an empty product $\prod_{w=0}^{-1}$ is taken to be one. Similarly, we obtain

$$[P(\lambda,\delta x)]^{\nu_\lambda} = \sum_{u=0}^{\nu_\lambda} \binom{\nu_\lambda}{u} (-1)^u \, e^{-u\delta x} \sum_{\{\xi_i:(0,\lambda-1);\Sigma=\nu_\lambda\}} \nu_\lambda! \prod_{g=0}^{\lambda-1} \frac{(\delta x)^{g\xi_g}}{\xi_g!(g!)^{\xi_g}} \cdot \qquad (2C\text{-}14)$$

Since

$$\prod_{i=0}^{N} z^{f_i} = z^{\sum_{i=0}^{N} f_i} \qquad (2C\text{-}15)$$

we can write (2C-13) and (2C-14), respectively, as

$$[P(j,x)]^{\mu_j} = \mu_j! \sum_{v=0}^{\mu_j} \binom{\mu_j}{v} (-1)^v \, e^{-vx} \sum_{\{\alpha_i:(0,j-1);\Sigma=\mu_j\}} \left(\sum_{w=0}^{j-1} \frac{1}{\alpha_w!(w!)^{\alpha_w}} \right)$$

$$\cdot \; x^{\sum_{w=0}^{j-1} w\alpha_w} \qquad (2C\text{-}16)$$

and

$$[P(\lambda,\delta x)]^{\nu_\lambda} = \nu_\lambda! \sum_{u=0}^{\nu_\lambda} \binom{\nu_\lambda}{u} (-1)^u e^{-u\delta x} \sum_{\{\xi_i:(0,\lambda-1);\Sigma=\nu_\lambda\}} \prod_{g=0}^{\lambda-1} \frac{1}{\xi_g!(g!)^{\xi_g}}$$

$$\cdot (\delta x)^{\sum_{g=0}^{\lambda-1} g\xi_g} \qquad (2C\text{-}17)$$

where empty sums are taken to be zero and empty products are taken to be one, as before.

We now combine (2C-9), (2C-10), (2C-11), (2C-16), and (2C-17) to obtain, after some rearrangement and factoring,

$$Q^{M-1} = \delta^{(L-\ell)(M-1)} \left(\frac{1}{\delta-1}\right)^{L(M-1)} \sum_{k=0}^{M-1} (M-1)! \sum_{\{\mu_i:(0,\ell);\Sigma=k\}} \sum_{\{\nu_i:(0,L-\ell);\Sigma=M-1-k\}}$$

$$\left\{ \prod_{j=0}^{\ell} \frac{(-1)^{j\mu_j}}{\mu_j!} \left[(1-\delta_{j,0}) \binom{L-\ell-1}{\ell-j} (\delta-1)^j \right]^{\mu_j} \right\}$$

$$\cdot \left\{ \prod_{\lambda=0}^{L-\ell} \frac{1}{\nu_\lambda!} \left[(1-\delta_{\lambda,0}) \binom{L-1}{L-\ell-\lambda} \left(\frac{\delta-1}{\delta}\right)^{\lambda} \right]^{\nu_\lambda} \right\}$$

$$\cdot \left(\sum_{\iota=0}^{\ell} \sum_{v_\iota=0}^{\mu_\iota} \right) \left(\sum_{\eta=0}^{L-\ell} \sum_{u_\eta=0}^{\nu_\eta} \right) \left\{ \prod_{\theta=0}^{\ell} \binom{\mu_\theta}{v_\theta} (-1)^{v_\theta} \mu_\theta! \right\} \left\{ \prod_{q=0}^{L-\ell} \binom{\nu_q}{u_q} (-1)^{u_q} \nu_q! \right\}$$

$$\cdot \exp\left[-\left(\sum_{\theta=0}^{\ell} v_\theta + \delta \sum_{q=0}^{L-\ell} u_q \right) x \right]$$

$$\cdot \left(\sum_{\tau=0}^{\ell} \sum_{\{\alpha_{\tau,i_\tau}:(0,\tau-1);\Sigma_\tau=\mu_\tau\}} \right) \left(\sum_{h=0}^{L-\ell} \sum_{\{\xi_{h,i_h}:(0,h-1);\Sigma_h=\nu_h\}} \right)$$

$$\left[\left(\prod_{\tau=0}^{\ell}\prod_{\theta_\tau=0}^{\tau}\prod_{w_{\theta_\tau}=0}^{\epsilon_\tau-1}\right)\frac{1}{\alpha_{\theta_\tau,w_{\theta_\tau}}!\,(w_{\theta_\tau}!)^{\alpha_{\theta_\tau,w_{\theta_\tau}}}}\right]$$

$$\left[\left(\prod_{h=0}^{L-\ell}\prod_{q_h=0}^{h-1}\prod_{g_{q_h}=0}^{q_h-1}\right)\frac{1}{\xi_{q_h,g_{q_h}}!\,(g_{q_h}!)^{\xi_{q_h,g_{q_h}}}}\right]$$

$$\cdot\left(\delta^{\sum_{h=0}^{L-\ell}\sum_{q_h=0}^{h}\sum_{g_{q_h}=0}^{q_h-1} g_{q_h}\ {}_{q_h,g_{q_h}}}\right)\ x^{\left(\sum_{\tau=0}^{\ell}\sum_{\theta_\tau=0}^{\tau}\sum_{w_{\theta_\tau}=0}^{\theta_\tau-1}\right)w_{\theta_\tau}\alpha_{\theta_\tau,w_{\theta_\tau}}+\left(\sum_{h=0}^{L-\ell}\sum_{q_h=0}^{h}\sum_{g_{q_h}=0}^{q_h-1}\right)g_{q_h}\ {}_{q_h,g_{q_h}}} \tag{2C-18}$$

where we have transformed products of sums (the sums coming from (2C-16) or (2C-17) for the powers of the incomplete gamma functions) into sums of products, with the accompanying proliferation of indices of summation, and have introduced the shorthand notations that, formally,

$$\left(\sum_{\iota=0}^{\ell}\sum_{v_\iota=0}^{\mu_\iota}\right)\triangleq\sum_{v_0=0}^{\mu_0}\sum_{v_1=0}^{\mu_1}\cdots\sum_{v_\ell=0}^{\mu_\ell}, \tag{2C-19a}$$

$$\left(\sum_{\tau=0}^{\ell}\sum_{\theta_\tau=0}^{\tau}\sum_{w_{\theta_\tau}=0}^{\theta_\tau}\right)\triangleq\left(\sum_{\theta_0=0}^{0}\sum_{w_{\theta_0}=0}^{\theta_0}\right)\left(\sum_{\theta_1=0}^{1}\sum_{w_{\theta_1}=0}^{\theta_1}\right)\cdots\left(\sum_{\theta_\ell=0}^{\ell}\sum_{w_{\theta_\ell}=0}^{\theta_\ell}\right) \tag{2C-19b}$$

with each part of the right-hand side of (2C-19b) being the notation defined in (2C-19a),

$$\left(\sum_{\tau=0}^{\ell}\ \sum_{\{\alpha_{\tau,i_\tau}:(0,\tau-1);\Sigma_\tau=\mu_\tau\}}\right)\triangleq\sum_{\{\alpha_{0,i_0}:(0,0);\Sigma_0=\mu_0\}}\cdots\sum_{\{\alpha_{\ell,i_\ell}:(0,\ell);\Sigma_\ell=\mu_\ell}} \tag{2C-19c}$$

where the individual summation symbols on the right-hand side of (2C-19c) are themselves the shorthand notation defined in (2C-6) with

$$\Sigma_\tau \triangleq \sum_{\tau=0}^{\ell} i_\tau \tag{2C-19d}$$

and

$$\left(\prod_{\tau=0}^{\ell}\prod_{\theta_\tau=0}^{\tau}\prod_{w_{\theta_\tau}=0}^{\theta_\tau}\right) \triangleq \left(\prod_{\theta_0=0}^{0}\prod_{w_{\theta_0}=0}^{\theta_0}\right)\cdot\left(\prod_{\theta_1=0}^{1}\prod_{w_{\theta_1}=0}^{\theta_1}\right)\cdots\left(\prod_{\theta_\ell=0}^{\ell}\prod_{w_{\theta_\ell}=0}^{\theta_\ell}\right) \tag{2C-20a}$$

with each term of the right-hand side of (2C-20a) defined by

$$\left(\prod_{\theta_\ell=0}^{\ell}\prod_{w_{\theta_\ell}=0}^{\theta_\ell}\right) \triangleq \prod_{w_0=0}^{0}\prod_{w_1=0}^{1}\cdots\prod_{w_\ell=0}^{\ell} . \tag{2C-20b}$$

Again, empty sums are interpreted as zero and empty products are interpreted as one.

We now return our attention to (2C-1). Writing Q^{M-1} for the quantity raised to the M-1 power and interchanging the order of summations and integration we have

$$\begin{aligned} P_s(e) = 1 - \sum_{\ell=0}^{L}\binom{L}{\ell}\gamma^\ell(1-\gamma)^{L-\ell}e^{-\ell\rho_T}e^{-(L-\ell)\rho_N}(-1)^{\ell M}\sum_{m=0}^{\infty}\frac{(\ell\rho_T)^m}{m!}\sum_{n=0}^{\infty}\frac{[(L-\ell)\rho_N]^n}{n!}\delta^{L-\ell+n} \\ \cdot\left[\sum_{r=0}^{\ell+m}(1-\delta_{r,0})(-1)^{m-r}\binom{L-\ell+n-1}{\ell+m-r}\left(\frac{1}{\delta-1}\right)^{L+m+n-r}\frac{1}{\Gamma(r)}\int_0^\infty e^{-x}x^{r-1}Q^{M-1}\,dx\right. \\ \left.+(-1)^m\sum_{r=0}^{L-\ell+n}(1-\delta_{r,0})\binom{\ell+m-1}{L-\ell+n-r}\left(\frac{1}{\delta-1}\right)^{L+m+n-r}\frac{1}{\Gamma(r)}\int_0^\infty e^{-\delta x}x^{r-1}Q^{M-1}\,dx\right] \end{aligned} \tag{2C-21}$$

where we have also made use of (2C-2). If we further interchange the order of integration with the multitude of summations embedded in Q^{M-1} as agiven by (2C-18) we have integrals of two forms to evaluate, namely

$$H_1 \triangleq \int_0^\infty x^{r-1+F+G}\,\delta^G \exp\left[-\left(1+\sum_{\theta=0}^{\ell} v_\theta + \delta\sum_{q=0}^{L-\ell} u_q\right)x\right] dx \qquad \text{(2C-22a)}$$

and

$$H_2 \triangleq \int_0^\infty x^{r-1+F+G}\,\delta^G \exp\left[-\left(\delta+\sum_{\theta=0}^{\ell} v_\theta + \delta\sum_{q=0}^{L-\ell} u_q\right)x\right] dx \qquad \text{(2C-22b)}$$

where

$$F \triangleq \sum_{\tau=0}^{\ell}\sum_{\theta_\tau=0}^{\tau}\sum_{w_{\theta_\tau}=0}^{\theta_\tau-1} w_{\theta_\tau}\alpha_{\theta_\tau, w_{\theta_\tau}} \qquad \text{(2C-23a)}$$

and

$$G \triangleq \sum_{h=0}^{L-\ell}\sum_{q_h=0}^{h}\sum_{g_{q_h}=0}^{q_h-1} g_{q_h}\xi_{q_h, g_{q_h}}. \qquad \text{(2C-23b)}$$

We can use (2C-8) to evaluate (2C-22) with the results

$$H_1 = \delta^G\,\Gamma(r+F+G)\left(1+\sum_{\theta=0}^{\ell} v_\theta + \delta\sum_{q=0}^{L-\ell} u_q\right)^{-r-F-G} \qquad \text{(2C-24a)}$$

and

$$H_2 = \delta^G\,\Gamma(r+F+G)\left(\delta+\sum_{\theta=0}^{\ell} v_\theta + \delta\sum_{q=0}^{L-\ell} u_q\right)^{-r-F-G}. \qquad \text{(2C-24b)}$$

Now we can write the final answer by substituting (2C-18) and (2C-24) into (2C-21). In order to simplify the final form of the answer, we will re-order

the summations, taking the sums over r to the innermost level, i.e. we have used the distributive property of multiplication over addition

$$(A+B)\sum_i \alpha_i = \sum_i \alpha_i (A+B). \qquad (2C-25)$$

Finally, then, we write $P_s(e)$ in a form having no integrals remaining to be evaluated:

$$P_s(e) = 1 - \sum_{\ell=0}^{L} \binom{L}{\ell} \gamma^{\ell}(1-\gamma)^{L-\ell} e^{-\ell\rho_T} e^{-(L-\ell)\rho_N} (-1)^{\ell M} \sum_{m=0}^{\infty} \frac{(\ell\rho_T)^m}{m!} \sum_{n=0}^{\infty} \frac{[(L-\ell)\rho_N]^n}{n!}$$

$$\cdot\ \delta^{LM-\ell M+n} \left(\frac{1}{\delta-1}\right)^{LM-L} \sum_{k=0}^{M-1} (M-1)! \sum_{\{\mu_i:(0,\ell);\Sigma=k\}} \ \sum_{\{\nu_i:(0,L-\ell);\Sigma=M-1-k\}}$$

$$\left\{\prod_{j=0}^{\ell} \frac{(-1)^{j\mu_j}}{\mu_j!} \left[(1-\delta_{j,0})\binom{L-\ell-1}{\ell-j}(\delta-1)^j\right]^{\mu_j}\right\}$$

$$\cdot \left\{\prod_{\lambda=0}^{L-\ell} \frac{1}{\nu_\lambda!} \left[(1-\delta_{\lambda,0})\binom{L-1}{L-\ell-\lambda}\left(\frac{\delta-1}{\delta}\right)^{\lambda}\right]^{\nu_\lambda}\right\}$$

$$\cdot \left(\sum_{\iota=0}^{\ell} \sum_{v_\iota=0}^{\mu_\iota}\right)\left(\sum_{\eta=0}^{L-\ell} \sum_{u_\eta=0}^{\nu_\eta}\right)\left\{\prod_{\theta=0}^{\ell}\binom{\mu_\theta}{v_\theta}(-1)^{v_\theta}\,\mu_\theta!\right\}\left\{\prod_{q=0}^{L-\ell}\binom{\nu_q}{u_q}(-1)^{u_q}\,\nu_q!\right\}$$

$$\cdot \left(\sum_{\tau=0}^{\ell} \sum_{\{\alpha_{\tau,i_\tau}:(0,\tau-1);\Sigma_\tau=\mu_\tau\}}\right)\left(\sum_{h=0}^{L-\ell} \sum_{\{\xi_{h,i_h}:(0,h-1);\Sigma_h=\nu_h\}}\right)$$

$$\left[\left(\prod_{\tau=0}^{\ell} \prod_{\theta_\tau=0}^{\tau} \prod_{w_{\theta_\tau}=0}^{\theta_\tau-1}\right) \frac{1}{\alpha_{\theta_\tau,w_{\theta_\tau}}!\,(w_{\theta_\tau}!)^{\alpha_{\theta_\tau,w_{\theta_\tau}}}}\right]$$

$$\left[\left(\prod_{h=0}^{L-\ell}\prod_{q_h=0}^{h}\prod_{g_{q_h}=0}^{q_h-1}\right)\frac{1}{\xi_{q_h,g_{q_h}}!\,(g_{q_h}!)^{\xi_{q_h,g_{q_h}}}}\right]$$

$$\cdot\ \delta^{\left(\sum\limits_{h=0}^{L-\ell}\sum\limits_{q_h=0}^{h}\sum\limits_{g_{q_h}=0}^{q_h-1}\right)g_{q_h}\xi_{q_h,g_{q_h}}}\left\{\sum_{r=0}^{\ell+m}(1-\delta_{r,0})(-1)^{m-r}\binom{L-\ell+n-1}{\ell+m-r}\left(\frac{1}{\delta-1}\right)^{L+m+n-r}\right.$$

$$\cdot\ \frac{1}{\Gamma(r)}\,\Gamma\left[r+\left(\sum_{\tau=0}^{\ell}\sum_{\theta_\tau=0}^{\tau}\sum_{w_{\theta_\tau}=0}^{\theta_\tau-1}\right)w_{\theta_\tau}\alpha_{\theta_\tau,w_{\theta_\tau}}+\left(\sum_{h=0}^{L-\ell}\sum_{q_h=0}^{h}\sum_{g_{q_h}=0}^{q_h-1}\right)g_{q_h}\xi_{q_h,g_{q_h}}\right]$$

$$\cdot\left(1+\sum_{\theta=0}^{\ell}v_\theta+\delta\sum_{q=0}^{L-\ell}u_q\right)^{-r-\left(\sum\limits_{\tau=0}^{\ell}\sum\limits_{\theta_\tau=0}^{\tau}\sum\limits_{w_{\theta_\tau}=0}^{\theta_\tau-1}\right)w_{\theta_\tau}\alpha_{\theta_\tau,w_{\theta_\tau}}-\left(\sum\limits_{h=0}^{L-\ell}\sum\limits_{q_h=0}^{h}\sum\limits_{g_{q_h}=0}^{q_h-1}\right)g_{q_h}\xi_{q_h,g_{q_h}}}$$

$$+\ (-1)^m\sum_{r=0}^{L-\ell+n}(1-\delta_{r,0})\binom{\ell+m-1}{L-\ell+n-r}\left(\frac{1}{\delta-1}\right)^{L+m+n-r}\frac{1}{\Gamma(r)}$$

$$\cdot\ \Gamma\left[r+\left(\sum_{\tau=0}^{\ell}\sum_{\theta_\tau=0}^{\tau}\sum_{w_{\theta_\tau}=0}^{\theta_\tau-1}\right)w_{\theta_\tau}\alpha_{\theta_\tau,w_{\theta_\tau}}+\left(\sum_{h=0}^{L-\ell}\sum_{q_h=0}^{h}\sum_{g_{q_h}=0}^{q_h-1}\right)g_{q_h}\xi_{q_h,g_{q_h}}\right]$$

$$\left.\cdot\left(\delta+\sum_{\theta=0}^{\ell}v_\theta+\delta\sum_{q=0}^{L-\ell}u_q\right)^{-r-\left(\sum\limits_{\tau=0}^{\ell}\sum\limits_{\theta_\tau=0}^{\tau}\sum\limits_{w_{\theta_\tau}=0}^{\theta_\tau-1}\right)w_{\theta_\tau}\alpha_{\theta_\tau,w_{\theta_\tau}}-\left(\sum\limits_{h=0}^{L-\ell}\sum\limits_{q_h=0}^{h}\sum\limits_{g_{q_h}=0}^{q_h-1}\right)g_{q_h}\xi_{q_h,g_{q_h}}}\right\}$$

(2C-26)

where empty sums are taken to be zero and empty products are taken to be one. The immensely complicated form in (2C-26) arises from the need to express explicitly the coefficients resulting from repeatedly taking powers of multinomials. Since

$$\delta = \sigma_T^2/\sigma_N^2 = \beta_N/\beta_T \tag{2C-27}$$

we can observe from (2C-27) that the error probability for MFSK has the form of a sum of exponentials of the signal-to-thermal-noise and signal-to-total-noise ratios weighted by rational functions of these two ratios. Other than allowing this observation, though, (2C-26) seems to be of little practical importance. However, it is worth noting that the result for the M-ary case belongs to the same class of functions (albeit a much more complicated member of the class) as the result for binary FSK.

The complicated form of (2C-26) arises through the coefficients in the rational function of β_N and β_T. It appears that a seemingly less complicated form is derivable using the J.C.P. Miller formula [3, p. 42] in lieu of (2C-5) to express powers of summations; however the approach would yield a reduced form of (2C-26), only to have it followed by coupled sets of recursive definitions of numerous coefficients. Overall, no meaningful reduction of the expression would be achieved.

Finally we note that (2C-26) gives the symbol error probability. The bit error probability is

$$P_b(e) = \frac{M}{2(M-1)} P_s(e) \tag{2C-28}$$

where $P_s(e)$ is given by (2C-26). We will not take the space here to write out $P_b(e)$ in full.

APPENDIX 2D

SPECIAL-CASE RESULTS OF ERROR RATE EXPRESSION FOR THE SQUARE-LAW LINEAR COMBINING RECEIVER DERIVED FROM THE RESULTS OF THE CHARACTERISTIC FUNCTION METHOD

It is of interest to examine several special cases of the general formulation for the error probability of the square-law linear combining receiver as derived using the characteristic function method. These special cases present simplified performance equations for several cases of practical interest.

2D.1 WIDEBAND JAMMING

In the case where the jammer fills the total system bandwidth ($\gamma=1$), the error rate expression becomes much simpler. When $\gamma=1$, only the term $\ell=L$ does not vanish in the outermost summation in (2-87). Then with $\ell=L$, only the term $n=0$ does not vanish in the summation over n. Therefore, with $\gamma=1$, (2-87) becomes

$$P_s(e|\gamma=1) = 1 - \int_0^\infty e^{-L\rho_T} \sum_{m=0}^{\infty} \frac{(L\rho_T)^m}{m!} \frac{1}{\Gamma(L+m)} e^{-x} x^{L+m-1} [P(L,x)]^{M-1} dx \tag{2D-1}$$

where we have also made use of the property

$$(0)_k = \begin{cases} 1, & k=0 \\ 0, & k\neq 0 \end{cases} \tag{2D-2}$$

to reduce the non-vanishing summations over r in (2-87) to a single term each. Since [4, eq. 9.6.10]

$$I_\nu(z) = \left(\frac{z}{2}\right)^\nu \sum_{k=0}^{\infty} \frac{(z^2/4)^k}{k!\,\Gamma(k+\nu+1)} \tag{2D-3}$$

we can replace the summation in (2D-1) by a modified Bessel function to obtain

$$P_s(e|\gamma=1) = 1 - \frac{e^{-L\rho_T}}{(L\rho_T)^{(L-1)/2}} \int_0^\infty x^{(L-1)/2}\, e^{-x}\, I_{L-1}\left(\sqrt{4L\rho_T x}\right) [P(L,x)]^{M-1}\, d\alpha. \tag{2D-4}$$

The integral in (2D-4) converges reasonably rapidly, making numerical evaluation of $P_s(e|\gamma=1)$ a straightforward task, However, an analytical form may also be obtained. For an integer first argument the incomplete gamma function $P(L,x)$ may be written as [4, eq. 6.5.13]

$$P(L,x) = 1 - e^{-x} \sum_{k=0}^{L-1} \frac{x^k}{k!}\,. \tag{2D-5}$$

Substitution of (2D-5) into (2D-4) yields

$$\begin{aligned} [P(L,x)]^{M-1} &= \left(1 - e^{-x} \sum_{k=0}^{L-1} \frac{x^k}{k!}\right)^{M-1} \\ &= \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m\, e^{-mx} \left(\sum_{k=0}^{L-1} \frac{x^k}{k!}\right)^m . \end{aligned} \tag{2D-6}$$

Now, the power of a summation in (2D-6) is a polynomial of the form

$$\left(\sum_{k=0}^{L-1} \frac{x^k}{k!}\right)^m = \sum_{k=0}^{m(L-1)} c_{m,k}\, x^k \tag{2D-7}$$

where the coefficients $c_{m,k}$ are related to the multinomial coefficients in a complicated fashion. We may consider $c_{m,k}$ to be defined by equating coefficients of like powers of x on both sides of (2D-7) or, alternatively, by a recursion relation obtained from the J.C.P. Miller formula [3, p. 42] as was done in Section 2.1 for the Gaussian channel. Using (2D-7) in (2D-6), we have

$$[P(L,x)]^{M-1} = \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m \sum_{k=0}^{m(L-1)} c_{m,k}\, x^k\, e^{-mx}. \tag{2D-8}$$

Substitution of (2D-8) into (2D-4) and interchanging the order of integration and summation yields

$$P_s(e|\gamma=1) = 1 - \frac{e^{-L\rho_T}}{(L\rho_T)^{(L-1)/2}} \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m \sum_{k=0}^{m(L-1)} c_{m,k} \int_0^\infty x^{k+(L-1)/2}\, e^{-x}$$

$$\cdot\; I_{L-1}(\sqrt{4L\rho_T x})\, dx. \tag{2D-9}$$

From [2, eq. 6.643.2 and 9.220.1], we have

$$\int_0^\infty x^{\mu-\frac{1}{2}}\, e^{-ax}\, I_{2\nu}(2\eta\sqrt{x})\, dx = \frac{\Gamma(\mu+\nu+\frac{1}{2})\, \eta^{2\nu}}{\Gamma(2\nu+1)\, \alpha^{\mu+\nu+\frac{1}{2}}}\; {}_1F_1(\nu+\mu+\tfrac{1}{2};2\nu+1;\eta^2/\alpha) \tag{2D-10}$$

which may be used to evaluate (2D-9) with the result

$$P_s(e|\gamma=1) = 1 - e^{-L\rho_T} \sum_{m=0}^{M-1} \binom{M-1}{m}(-1)^m \sum_{k=0}^{m(L-1)} c_{m,k}(L)_k\, \frac{1}{m^{k+L}}\; {}_1F_1(k+L;L;L\rho_T/m) \tag{2D-11}$$

where the coefficients $c_{m,k}$ are defined implicitly by (2D-7). Equation (2D-1) agrees with the results obtained in Section 2.2.

2D.2 ONE HOP PER BIT

The case of one hop per bit (L=1) corresponds to MFSK without frequency hopping. If we set L=1 and explicitly write out the two terms of the summation over ℓ in (2-87), taking into account the degenerate sums which arise when $\ell=0$ and $\ell=M$, and considering the value of $(0)_k$ given by (2D-2), we have, after some rearrangement,

$$P_s(e|L=1) = 1 - \delta(1-\gamma)e^{-\rho_N} \int_0^\infty \sum_{n=0}^{\infty} \frac{(\delta x \rho_N)^n}{n!\Gamma(n+1)} e^{-\delta x} [P(1,\delta x)]^{M-1}\, dx$$

$$- \gamma\, e^{-\rho_T} \int_0^\infty \sum_{m=0}^{\infty} \frac{(x/\rho_T)^m}{m!\Gamma(m+1)}\, e^{-x}\, [P(1,x)]^{M-1}\, dx. \qquad (2D\text{-}12)$$

Using (2D-3), we can write (2D-12) as

$$P_s(e|L=1) = 1 - \delta(1-\gamma)e^{-\rho_N} \int_0^\infty e^{-\delta x}\, I_0(2\sqrt{\delta\rho_N x})\, [P(1,\delta x)]^{M-1}\, dx$$

$$- \gamma\, e^{-\rho_T} \int_0^\infty e^{-x}\, I_0(2\sqrt{\rho_T x})\, [P(1,x)]^{M-1}\, dx. \qquad (2D\text{-}13)$$

From (2D-15)

$$P(1,x) = 1 - e^{-x} \qquad (2D\text{-}14)$$

and thus, with the aid of the binomial theorem, (2D-13) reduces to

$$P_s(e|L=1) = 1 - \delta(1-\gamma)\, e^{-\rho_N} \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m \int_0^\infty e^{-(m+1)\delta x}\, I_0(\sqrt{4\delta\rho_N x})\, dx$$

$$- \gamma\, e^{-\rho_T} \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m \int_0^\infty e^{-(m+1)x}\, I_0(\sqrt{4\rho_T x})\, dx. \quad (2D\text{-}15)$$

The integrals in (2D-15) may be evaluated using (2D-10) with $\mu=\frac{1}{2}$, $\nu=0$. The result is

$$P_s(e|L=1) = 1 - (1-\gamma)\, e^{-\rho_N} \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m \frac{1}{m+1}\, {}_1F_1\left(1;1;\frac{\rho_N}{m+1}\right)$$

$$- \gamma\, e^{-\rho_T} \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m \frac{1}{m+1}\, {}_1F_1\left(1;1;\frac{\rho_T}{m+1}\right). \qquad \text{(2D-16)}$$

Since [4, 13.6.12]

$$ {}_1F_1(a;a;z) = e^z, \qquad \text{(2D-17)}$$

(2D-16) becomes

$$P_s(e|L=1) = 1 - (1-\gamma)\, e^{-\rho_N} \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m \frac{1}{m+1} \exp\left(\frac{\rho_N}{m+1}\right)$$

$$- \gamma\, e^{-\rho_T} \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m \frac{1}{m+1} \exp\left(\frac{\rho_T}{m+1}\right). \qquad \text{(2D-18)}$$

We now observe that the m=0 terms of the two sums in (2D-18) add up to 1. Thus, we may further simplify the result to

$$P_s(e|L=1) = \sum_{m=1}^{M-1} \binom{M-1}{m} (-1)^{m+1} \frac{1}{m+1} \left[(1-\gamma) \exp\left(-\frac{m\rho_N}{m+1}\right) + \gamma \exp\left(-\frac{m\rho_T}{m+1}\right)\right] \qquad \text{(2D-19)}$$

where we have combined the two summations into one and factored out the common coefficients. The result in (2D-19) is readily identifiable as the weighted sum of the result for MFSK given by Stein and Jones [17, eq. 14-45] which results from averaging over two possible noise states (jammed and unjammed). If we further specialize (2D-19) to the case of M=2 (binary), we obtain the result of our earlier work on BFSK [6, eq. 2.3-38], namely

$$P_b(e|L=1,M=2) = \tfrac{1}{2}(1-\gamma)\, e^{-\rho_N/2} + \tfrac{1}{2}\gamma\, e^{-\rho_T/2}. \qquad \text{(2D-20)}$$

APPENDIX 2E

PROBABILITY DENSITY FUNCTION FOR THE WEIGHTED SUM OF TWO INDEPENDENT NONCENTRAL CHI-SQUARED RANDOM VARIABLES

Let x and y be two independent noncentral chi-square random variables,

$$x \sim \chi^2(2n;\lambda_1) \tag{2E-1a}$$

and

$$y \sim \chi^2(2m;\lambda_2). \tag{2E-1b}$$

We shall derive the probability density function for the random variable

$$u = x + Ky. \tag{2E-2}$$

The joint probability density function of x and y is

$$p_1(x,y) = \frac{1}{4}\exp\left[-\frac{1}{2}(x+y+\lambda_1+\lambda_2)\right]\sum_{k=0}^{\infty}\frac{(\lambda_1/2)^k\,(x/2)^{k+n-1}}{k!(k+n-1)!}\sum_{r=0}^{\infty}\frac{(\lambda_2/2)^r(y/2)^{r+m-1}}{r!(r+m-1)},$$

$$x \geq 0, \quad y \geq 0. \tag{2E-3}$$

We define a transformation of variables

$$\left.\begin{array}{l} u = x + Ky \\ v = x - Ky \end{array}\right\} \quad u > 0, \quad |v| < u \tag{2E-4a}$$

or

$$\left.\begin{array}{l} x = \frac{1}{2}(u+v) \\ y = \frac{1}{2K}(u-v) \end{array}\right\} \quad u > 0, \quad |v| < u. \tag{2E-4b}$$

The joint probability density function of the new variables u and v is

$$p_2(u,v) = \frac{1}{2K}\, p_1\left(\frac{u+v}{2}\,,\ \frac{u-v}{2}\right)$$

$$= \frac{1}{8K}\exp\left(-\,\frac{\lambda_1+\lambda_2}{2} + \frac{u+v}{4} - \frac{u-v}{4}\right)$$

$$\cdot \sum_{k=0}^{\infty}\sum_{r=0}^{\infty} \frac{(\lambda_1/2)^k\ (\lambda_2/2)^r}{k!r!}\ \frac{\left(\frac{u+v}{4}\right)^{k+n-1}\left(\frac{u-v}{4K}\right)^{r+m-1}}{(k+n-1)!\ (r+m-1)!}\,,$$

$$u > 0,\quad |v| < u. \tag{2E-5}$$

To integrate out v, we use the integral

$$\int_{-u}^{u} \left(\frac{u+v}{4}\right)^{k+m-1}\left(\frac{u-v}{4}\right)^{r+m-1} e^{-bv}dv = \left(\frac{u}{4}\right)^{k+r+n+m-2}\left(\frac{1}{K}\right)^{r+m-1}\int_{-u}^{u}\left(1+\frac{v}{u}\right)^{k+n-1}\left(1-\frac{v}{u}\right)^{r+m+1} e^{-bv}\,dv$$

$$= u\left(\frac{u}{4}\right)^{k+r+n+m-2}\left(\frac{1}{K}\right)^{r+m+1}\int_{-1}^{1}(1+x)^{k+n-1}(1-x)^{r+m-1}e^{-bux}dx. \tag{2E-6}$$

From [4, eq. 13.2.2],

$$\int_{-1}^{1}(1+x)^{k+n-1}\ (1-x)^{r+m-1}\ e^{-bux}\ dx =$$

$$e^{-bu}\ 2^{k+n+r+m-1}\ \frac{\Gamma(k+n)\Gamma(r+m)}{\Gamma(k+n+r+m)}\ {}_1F_1(r+m;\ k+n+r+m;\ 2bu). \tag{2E-7}$$

Substituting (2E-7) and (2E-6) into (2E-5) and using $b = \frac{K-1}{4K}$ gives the final probability density function

$$p_3(u) = \frac{1}{2K^m}\exp\left(-\,\frac{u+\lambda_1+\lambda_2}{2}\right)\sum_{k=0}^{\infty}\sum_{r=0}^{\infty}\frac{\left(\frac{\lambda_1}{2}\right)^k\left(\frac{\lambda_2}{2K}\right)^r\left(\frac{u}{2}\right)^{k+r+n+m-1}}{k!\ r!\ (k+r+n+m-1)!}\ {}_1F_1\left(r+m;\ k+r+m+n;\ \frac{K-1}{K}\,\frac{u}{2}\right). \tag{2E-8}$$

APPENDIX 2F

BIT ERROR PROBABILITY FOR THE SQUARE-LAW LINEAR COMBINING RECEIVER FOR THE SPECIAL CASE OF L=2 HOPS PER SYMBOL

For L=2 hops/symbol, (2-55) is the sum of three terms. In the derivation below, we first compute each term of (2-55) and then combine these results to obtain the final special-case equation for L=2.

For $\ell=0$ we have the case of no jamming on either hop; (2-26b) becomes

$$p_{Z_2}(\zeta_2) = \frac{1}{2\sigma_N^2}\left(\frac{\zeta_2}{2\sigma_N^2}\right)\exp\left(-\frac{\zeta_2}{2\sigma_N^2}\right), \quad \zeta_2 > 0; \tag{2F-1}$$

and (2-60) is expressible in closed form as

$$p(z_i|z_1 = \zeta_1;0) = 1 - e^{-\zeta_1/2\sigma_N^2}\left(1 + \frac{\zeta_1}{2\sigma_N^2}\right). \tag{2F-2}$$

Also, for the signal channel, (2-73) with $\ell=0$ becomes

$$p_1(\zeta_1|0) = \frac{1}{2\sigma_N^2}\left(\frac{\zeta_1}{8\sigma_N^2\rho_N}\right)^{\frac{1}{2}} I_1\left(\sqrt{\frac{8\rho_N\zeta_1}{\sigma_N^2}}\right). \tag{2F-3}$$

If we expand the modified Bessel function in (2F-3) in a Taylor series, substitute (2F-1) and (2F-3) into (2-61), make the change of variable $\alpha = \zeta_1/\sigma_N^2$, and interchange the order of summation and integration, we obtain

$$P_s(e|0) = 1 - \frac{1}{2} e^{-\lambda_0/2} \sum_{k=0}^{\infty} \frac{1}{k!(k+1)!} \left(\frac{\lambda_0}{2}\right)^k \int_0^{\infty} e^{-\alpha/2} \left(\frac{\alpha}{2}\right)^{k+1}$$

$$\cdot \left[1 - e^{-\alpha/2} \left(1 + \frac{\alpha}{2}\right)\right]^{M-1} d\alpha \tag{2F-4}$$

where

$$\lambda_0 \triangleq 2\rho_N. \tag{2F-5}$$

Applying the binomial theorem twice to (2F-4) yields

$$P_s(e|0) = 1 - \frac{1}{2} e^{-\lambda_0/2} \sum_{k=0}^{\infty} \frac{1}{k!(k+1)!} \left(\frac{\lambda_0}{2}\right)^k \sum_{m=0}^{M-1} \binom{M-1}{m} (-1)^m \sum_{p=0}^{m} \binom{m}{p}$$

$$\cdot \int_0^{\infty} \left(\frac{\alpha}{2}\right)^{k+p+1} e^{-(m+1)\alpha/2} \, d\alpha. \tag{2F-6}$$

The integral in (2F-6) is equal to

$$2\left(\frac{1}{m+1}\right)^{k+p+2} (k+p+1)!.$$

Upon interchanging the order of summations, we obtain

$$P_s(e|0) = 1 - e^{-\lambda_0/2} \sum_{m=0}^{M} \binom{M-1}{m} \frac{(-1)^m}{(m+1)^2} \sum_{p=0}^{m} \binom{m}{p} \frac{(p+1)!}{(m+1)^p} \sum_{k=0}^{\infty} \frac{(p+2)_k}{(2)_k k!} \left[\frac{\lambda_0}{2(m+1)}\right]^k. \tag{2F-7}$$

The summation over k in (2F-7) is a confluent hypergeometric function. Upon applying Kummer's Transformation it is recognized as the generalized Laguerre polynomial [4, 13.6.9] and hence (2F-7) becomes

$$P_s(e|0) = 1 - \sum_{m=0}^{M-1} \binom{M-1}{m} \frac{(-1)^m}{(m+1)^2} \sum_{p=0}^{m} \binom{m}{p} \frac{p!}{(m+1)^p} \exp\left(-\frac{2m\rho_N}{m+1}\right) \mathcal{L}_p^1\left(-\frac{2\rho_N}{m+1}\right). \tag{2F-8}$$

For $\ell=2$, the result is obtained from (2F-8) by replacing ρ_N by ρ_T.

For $\ell=1$, a similar set of manipulations on (2-62), (2-72), (2-73), and (2-61) yields

$$P_s(e|1) = 1 - \frac{1}{2\delta} e^{-\lambda_1/2} \sum_{k=0}^{\infty} \frac{1}{k!}\left(\frac{\lambda_{0,1}}{2}\right)^k \sum_{r=0}^{k} \binom{k}{r}\left(\frac{\lambda_{1,1}}{\lambda_{0,1}}\right) \frac{\delta^{-r}}{(k+1)!}$$

$$\cdot \int_0^{\infty} e^{-\alpha/2} \left(\frac{\alpha}{2}\right)^{k+1} \left[1 - \frac{1}{\delta-1}\left(\delta e^{-\alpha/2\delta} - e^{-\alpha/2}\right)\right]^{M-1} {}_1F_1\left(r+1;\ k+2;\ \frac{(\delta-1)\alpha}{2\delta}\right) d\alpha \tag{2F-9}$$

where $\lambda_{0,1} \triangleq 2\rho_N$, $\lambda_{1,1} \triangleq 2\rho_T$, $\lambda_1 \triangleq \lambda_{0,1} + \lambda_{1,1}$ and we have used [18, eq. A.1.19] to evaluate (2-60) for $\ell=1$. To evaluate the integral in (2F-9) we expand the (M-1)-st power using the binomial theorem and use [2, eq. 7.621.5] to obtain

$$P_s(e|1) = 1 - \sum_{m=0}^{M-1} \binom{M-1}{m} \frac{(-1)^m}{(\delta-1)^m} \sum_{p=0}^{m} \binom{m}{p} (-1)^p \delta^{m-p} \frac{\delta}{[(\delta-1)p+m+\delta]\ [(\delta-1)p+m+1]}$$

$$\cdot \exp\left\{-\rho_N\left[\frac{(\delta-1)p+m}{(\delta-1)p+m+\delta}\right] - \rho_T\left[\frac{(\delta-1)p+m}{(\delta-1)p+m+1}\right]\right\}. \tag{2F-10}$$

Finally, then, putting (2F-8) and (2F-10) into (2-55) yields, with a bit of algebraic simplification, the desired special case equation

$$P_b(e) = \frac{M}{2(M-1)} \sum_{m=1}^{M-1} \binom{M-1}{m}(-1)^{m+1} \sum_{p=0}^{m} \binom{m}{p} \left\{\frac{p!}{(m+1)^{2p}} \left[(1-\gamma)^2 \exp\left(-\frac{2m\rho_N}{m+1}\right) \mathcal{L}_p^1\left(\frac{-2\rho_N}{m+1}\right)\right.\right.$$

$$\left. + \gamma^2 \exp\left(-\frac{2m\rho_T}{m+1}\right) \mathcal{L}_p^1\left(-\frac{2\rho_T}{m+1}\right)\right] + 2\gamma(1-\gamma) \left(\frac{\delta}{\delta-1}\right)^m \left(-\frac{1}{\delta}\right)^p$$

$$\left. \cdot \frac{\delta}{[(\delta-1)p+m+\delta]\ [(\delta-1)p+m+1]} \exp\left\{-\rho_N\left[\frac{(\delta-1)p+m}{(\delta-1)p+m+\delta}\right] - \rho_T\left[\frac{(\delta-1)p+m}{(\delta-1)p+m+1}\right]\right\}\right\}. \tag{2F-11}$$

APPENDIX 2G
COMPUTER PROGRAM TO COMPUTE BIT ERROR PROBABILITY FOR SQUARE-LAW LINEAR COMBINING RECEIVER IN THE PARTIAL-BAND NOISE-JAMMING CHANNEL USING NUMERICAL INTEGRATION OF (2-61)

The following pages contain a listing of the FORTRAN-77 program used to obtain numerical values of the probability of error for the square-law linear combining receiver in the presence of partial-band noise jamming by means of two-dimensional numerical integration.

The subroutines DQATR and DQATR2 are identical (except for the name) numerical integration routines using the Romberg method. There were obtained by converting the subroutine QATR from the Digital Equipment Corporation Scientific Subroutine Package [19] to double precision. Two copies are used to avoid recursion when performing the two-dimensional integration.

For a listing of subprogram DBINCO, see Appendix 4F, listing page 8; for DXBESI, Appendix 4G, listing pages 12-13; and for DXI, Appendix 4I.

```
0001          PROGRAM MFSKFH
        C-------------------------------------------------------------
        C THIS PROGRAM COMPUTES THE EXACT EQUATION FOR THE BIT ERROR
        C PROBABILITY FOR M-ARY FSK/FH WITH L HOPS/SYMBOL IN THE PRESENCE
        C OF PARTIAL BAND NOISE JAMMING AND THERMAL NOISE.
        C
        C COMPUTATIONS ARE DONE BY NUMERICAL INTEGRATION OF THE PROBABILITY
        C DENSITY OF THE EVENT OF MAKING A CORRECT DECISISON.  THEN,
        C                P(ERROR) = 1 - P(CORRECT)
        C ON A SYMBOL BASIS.
        C     THE INTEGRAL FOR P(CORRECT) IS THEN COMBINED WITH THE
        C     1 = INTEGRAL OF PDF OF SIGNAL CHANNEL
        C     TO YIELD P(ERROR) = INTEGRAL OF (PDF OF SIG. CH.) * (1-(PDF)**M)
        C     WHERE (PDF) IS THE PDF OF A NON-SIGNAL CHANNEL
        C THE CONVERSION TO BIT ERROR PROBABILITY ASSUMES
        C ORTHOGONAL SIGNALS, IN WHICH CASE:
        C
        C       P(BIT ERROR) = (M/(2M-2)) * P(SYMBOL ERROR)
        C-------------------------------------------------------------
        C PROGRAMMER: ROBERT H. FRENCH
        C              7 DEC 82
        C
        C VERSION 8.3.1
        C       OVERFLOW FIXES IN GP -- 16 NOV 82, RHF.
        C
0002          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003          BYTE REPLY,YES,NO,BLANK
0004          COMMON /INPUTS/ MLIST(5), LLIST(10), DEBNOL(5), DEBNJL(11),
            $ GAMLST(31)
0005          COMMON /OUTPUT/ PB(11)
0006          COMMON /SIZE/ NM, NL, NO, NJ, NG
0007          COMMON /PARMS/ GAMMA, DELTA, OODMO, MARY, FM, LL, FLL,
            $EHNO, EHNT, DELTAL, OODMOL, DMO
0008          COMMON /DELCON/ DC1, DC2, DC2I, DC3
0009          DATA YES,NO/'Y','N'/,BLANK/' '/
0010    1001  WRITE(5,1000)
0011    1000  FORMAT(' DO YOU WANT UNDERFLOW MESSAGES? [N] ',$)
0012          READ(5,1002)REPLY
0013    1002  FORMAT(A1)
0014          IF(REPLY.EQ.BLANK)REPLY=NO
0015          IF(REPLY.NE.YES.AND.REPLY.NE.NO)GOTO 1001
0016          IF(REPLY.EQ.YES)
            $   CALL ERRSET(74,.TRUE.,.FALSE.,.FALSE.,.TRUE.,255)
0017    1901  WRITE(5,1900)
0018    1900  FORMAT(' MAKE OVERFLOWS FATAL? [N] ',$)
0019          READ(5,1002)REPLY
0020          IF(REPLY.EQ.BLANK)REPLY=NO
0021          IF(REPLY.NE.YES.AND.REPLY.NE.NO)GOTO 1901
0022          IF(REPLY.EQ.YES)
            $   CALL ERRSET(72,.FALSE.,.TRUE.,.FALSE.,.TRUE.,15)
        C READ INPUT FROM TI:
0023          CALL GET
```



```
        C START TIMING THINGS AFTER INTERACTIVE INPUT....
0024          RTIME=SECNDS(0.)
        C LOOP ON M (ORDER OF ALPHABET)
0025          DO 900 IM=1,NM
0026          MARY=MLIST(IM)
0027          FM=MARY
0028          FK=DLOG2(FM)
0029          W2B=FM/(2.DO*(FM-1.DO))
        C LOOP ON HOPS/SYMBOL
0030          DO 800 IL=1,NL
0031          LL=LLIST(IL)
0032          FLL=LL
        C LOOP ON EB/NO
0033          DO 700 IO=1,NO
0034          EBNO=UNDB(DEBNOL(IO))
0035          ESNO=EBNO*FK
0036          EHNO=ESNO/FLL
0037          CALL PUT1(IM,IL,IO)
        C LOOP ON GAMMA
0038          DO 600 IG=1,NG
0039          GAMMA=GAMLST(IG)
        C LOOP ON EB/NJ
0040          DO 500 IJ=1,NJ
0041          EBNJ=UNDB(DEBNJL(IJ))
0042          ESNJ=EBNJ*FK
0043          EHNJ=ESNJ/FLL
0044          EHNT=GAMMA/(GAMMA/EHNO + 1.DO/EHNJ)
0045          DELTA=1.DO+EHNO/(GAMMA*EHNJ)
0046          DMO=DELTA-1.DO
0047          DELTAL=DLOG(DELTA)
0048          OODMO=1.DO/(DELTA-1.DO)
0049          OODMOL=DLOG(OODMO)
0050          DC2=DELTA*OODMO
0051          DC2I=1.DO/DC2
0052          DC3=1.DO-DELTA
0053          CALL PRERR(PWE)
0054          PB(IJ)=W2B*PWE
0055    500   CONTINUE
0056          CALL PUT2(IG)
0057    600   CONTINUE
0058    700   CONTINUE
0059    800   CONTINUE
0060    900   CONTINUE
0061          RTIME=SECNDS(RTIME)
0062          WRITE(6,1)RTIME
0063    1     FORMAT(///' ELAPSED WALL-CLOCK TIME = ',F10.2,' SECONDS.')
0064          STOP 0
0065          END
```

```
          SUBROUTINE GET
C SUBROUTINE TO READ INPUT PARAMETERS INTERACTIVELY
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          DIMENSION GAMDFL(31)
          LOGICAL*1 EJECT
          BYTE REPLY,YES,NO,BLANK
          COMMON /EJECT1/ EJECT
          COMMON /INPUTS/ MLIST(5), LLIST(10), DEBNOL(5), DEBNJL(11),
     $ GAMLST(31)
          COMMON /SIZE/ NM, NL, NO, NJ, NG
          DATA YES, NO, BLANK /'Y', 'N', ' '/
          DATA GAMDFL/
     $ 1.D-3,1.5D-3,2.D-3,3.D-3,4.D-3,5.D-3,6.D-3,7.D-3,8.D-3,9.D-3,
     $ 1.D-2,1.5D-2,2.D-2,3.D-2,4.D-2,5.D-2,6.D-2,7.D-2,8.D-2,9.D-2,
     $ 1.D-1,1.5D-1,2.D-1,3.D-1,4.D-1,5.D-1,6.D-1,7.D-1,8.D-1,9.D-1,
     $ 1.D0 /
          WRITE(5,1)
1         FORMAT(' 0 FOR NUMBER OF VALUES TAKES DEFAULTS IN [ ]')
100       WRITE(5,2)
2         FORMAT(' HOW MANY VALUES OF M? [M=2 ONLY] ',$)
          READ(5,3,ERR=100)NM
3         FORMAT(I2)
          IF(NM.LT.0.OR.NM.GT.5)GOTO 100
          IF(NM.EQ.0)GOTO 103
          DO 102 IN=1,NM
101       WRITE(5,4)IN
4         FORMAT(5X,'M(',I1,') = ',$)
          READ(5,5,ERR=101)MLIST(IN)
5         FORMAT(I5)
          IF(MLIST(IN).LE.1)GOTO 101
102       CONTINUE
          GOTO 104
103       NM=1
          MLIST(1)=2
104       WRITE(5,6)
6         FORMAT(' HOW MANY VALUES OF L? [L=2 ONLY] ',$)
          READ(5,3,ERR=104)NL
          IF(NL.LT.0.OR.NL.GT.10)GOTO 104
          IF(NL.EQ.0)GOTO 107
          DO 106 IN=1,NL
105       WRITE(5,7)IN
7         FORMAT(5X,'L(',I2,') = ',$)
          READ(5,5,ERR=105)LLIST(IN)
          IF(LLIST(IN).LE.0)GOTO 105
106       CONTINUE
          GOTO 108
107       NL=1
          LLIST(1)=2
108       WRITE(5,8)
8         FORMAT(' HOW MANY VALUES OF EB/NO? [13.35 DB ONLY] ',$)
          READ(5,3,ERR=108)NO
```



```
          IF(NO.LT.0.OR.NO.GT.5)GOTO 108
          IF(NO.EQ.0)GOTO 111
          DO 110 IN=1,NO
109       WRITE(5,9)IN
9         FORMAT(5X,'EB/NO(',I1,') (DB) = ',$)
          READ(5,10,ERR=109)DEBNOL(IN)
10        FORMAT(F5.2)
110       CONTINUE
          GOTO 112
111       NO=1
          DEBNOL(1)=13.35
112       WRITE(5,11)
11        FORMAT(' HOW MANY VALUES OF EB/NJ? [0(5)50 DB] ',$)
          READ(5,5,ERR=112)NJ
          IF(NJ.LT.0.OR.NJ.GT.11)GOTO 112
          IF(NJ.EQ.0)GOTO 115
          DO 114 IN=1,NJ
113       WRITE(5,12)IN
12        FORMAT(5X,'EB/NJ(',I2,') (DB) = ',$)
          READ(5,10,ERR=113)DEBNJL(IN)
114       CONTINUE
          GOTO 117
115       NJ=11
          DO 116 IN=1,NJ
          DEBNJL(IN)=5*(IN-1)
116       CONTINUE
117       WRITE(5,13)
13        FORMAT(' HOW MANY VALUES OF GAMMA? [.001 TO 1 IN 31 STEPS] ',$)
          READ(5,3,ERR=117)NG
          IF(NG.LT.0.OR.NG.GT.31)GOTO 117
          IF(NG.EQ.0)GOTO 120
          DO 119 IN=1,NG
118       WRITE(5,14)IN
14        FORMAT(5X,'GAMMA(',I2,') = ',$)
          READ(5,15,ERR=118)GAMLST(IN)
15        FORMAT(D10.3)
          IF(GAMLST(IN).LE.0.D0.OR.GAMLST(IN).GT.1.D0)GOTO 118
119       CONTINUE
          GOTO 122
120       NG=31
          DO 121 IN=1,NG
          GAMLST(IN)=GAMDFL(IN)
121       CONTINUE
122       WRITE(5,16)
16        FORMAT(' SUPPRESS PAGE EJECTS ON PRINT-OUT? [Y] ',$)
          READ(5,17)REPLY
17        FORMAT(A1)
          IF(REPLY.NE.YES.AND.REPLY.NE.NO.AND.REPLY.NE.BLANK)GOTO 122
          IF(REPLY.EQ.BLANK.OR.REPLY.EQ.YES)EJECT=.FALSE.
          IF(REPLY.EQ.NO)EJECT=.TRUE.
          RETURN
          END
```

```
PDP-11 FORTRAN-77 V4.0-1        10:09:12    27-Feb-84           Page 5

              SUBROUTINE PUT1(IM,IL,IO)
      C
      C SUBROUTINE TO OUTPUT PAGE HEADERS
      C
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              LOGICAL*1 EJECT
              COMMON /EJECT1/ EJECT
              COMMON /INPUTS/ MLIST(5), LLIST(10), DEBNOL(5), DEBNJL(11),
          $ GAMLST(31)
              COMMON /SIZE/ NM, NL, NO, NJ, NG
              IF(EJECT)WRITE(6,999)
      999     FORMAT('1')
              WRITE(6,1)MLIST(IM),LLIST(IL)
      1       FORMAT(39X,'EXACT ANALYSIS FOR ',I2,'-ARY FSK/FH WITH L = ',
          $ I2,' HOPS/SYMBOL'/' USING SPECIAL CASE EQUATION WITH SOME',
          $ ' INTEGRALS EVALUATED ANALYTICALLY, ONE DONE NUMERICALLY'//)
              WRITE(6,2)DEBNOL(IO)
      2       FORMAT(1X,'EB/NO = ',F6.2,' DB'//)
              WRITE(6,3)(DEBNJL(IOUT),IOUT=1,NJ)
      3       FORMAT(11X,<NJ>(' EB/NJ (DB)')/9X,<NJ>(5X,F6.2)/12X,
          $ <11*NJ-1>('-')/4X,'GAMMA',2X,<NJ>(5X,'PB',4X))
              RETURN
              END

PDP-11 FORTRAN-77 V4.0-1        10:09:16    27-Feb-84           Page 6

              SUBROUTINE PUT2(IG)
      C
      C WRITE ONE LINE OF RESULTS
      C
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              COMMON /INPUTS/ MLIST(5), LLIST(10), DEBNOL(5), DEBNJL(11),
          $ GAMLST(31)
              COMMON /OUTPUT/ PB(11)
              COMMON /SIZE/ NM, NL, NO, NJ, NG
              WRITE(6,1)GAMLST(IG),(PB(IOUT),IOUT=1,NJ)
      1       FORMAT(1X,1PD10.3,1P<NJ>D11.3)
              RETURN
              END

PDP-11 FORTRAN-77 V4.0-1        10:09:18    27-Feb-84           Page 7

              DOUBLE PRECISION FUNCTION UNDB(DB)
      C
      C FUNCTION TO CONVERT FROM DECIBELS TO NUMERIC RATIO
      C
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              UNDB=10.D0**(DB/10.D0)
              RETURN
              END

PDP-11 FORTRAN-77 V4.0-1        10:09:19    27-Feb-84           Page 8

              DOUBLE PRECISION FUNCTION DLOG2(X)
      C
      C DOUBLE PRECISION LOGARITHM TO BASE 2
      C
      C DLOG2(X) = DLOG(X)/DLOG(2)
      C
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              DLOG2=DLOG(X)/0.69314718055994530941723210D0
              RETURN
              END

PDP-11 FORTRAN-77 V4.0-1        10:09:20    27-Feb-84           Page 9

              DOUBLE PRECISION FUNCTION DXF(D,F)
      C
      C FUNCTION TO ALLOW RAISING NON-NEGATIVE DOUBLE PRECISION
      C TO DOUBLE PRECISION POWER
      C
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              IF(D)10,20,30
      10      WRITE(5,11)
      11      FORMAT(' ERROR--NEGATIVE D.P. ** D.P., EXEC. TERMINATED. (DXF)')
              STOP 9999
      20      DXF=0.D0
              IF(F.EQ.0.D0)DXF=1.D0
              RETURN
      30      DXF=D**F
              RETURN
              END
```

```
            SUBROUTINE PRERR(PWE)
      C
      C SUBROUTINE TO COMPUTE PROBABILITY OF CORRECT WORD DECISION
      C
            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
            COMMON /PARMS/ GAMMA, DELTA, OODMO, MARY, FM, LL, FLL,
           $EHNO, EHNT, DELTAL, OODMOL, DMO
            COMMON /INDEX/ L, LLL, LP1
            COMMON /POWERS/ XM, XN, EXPM, EXPN
            COMMON /BINCS/ BLL1L1, BLL1L
            COMMON /DELCON/  DC1, DC2, DC2I, DC3
            OMG=1.D0-GAMMA
            SUMLMN=0.D0
            LLP1=LL+1
            LP1I=1
            DO 900 LP1=LP1I,LLP1
            L=LP1-1
            DC1=DC2**LL/DELTA**L
            BLL1L1=DBINCO(LL-1,L-1)
            BLL1L=DBINCO(LL-1,L)
            LLL=LL-L
            FL=L
            IF(L.GT.0)GOTO 901
            BCL=1.D0
            GOTO 902
      901   BCL=BCL*((FLL-FL+1.D0)/FL)
      902   PARTL=BCL*(GAMMA**L) * DXI(OMG,LL-L)
            IF(PARTL.EQ.0.D0)GOTO 900
            XM=FL*EHNT
            XN=(FLL-FL)*EHNO
            EXPM=DEXP(-XM)
            EXPN=DEXP(-XN)
            CALL DOINT(ANSWER)
            TERML=ANSWER*PARTL
            SUMLMN=SUMLMN + TERML
      900   CONTINUE
            PWE=SUMLMN
            RETURN
            END
```



```
            SUBROUTINE DOINT(RESULT)
      C
      C THIS SUBROUTINE PERFORMS THE INTEGRAL
      C
      C             INFINITY
      C               -
      C  RESULT =   /   GRAND(X) DX
      C            -
      C            0
      C
      C VERSION 2.1.0
      C  IDENTIFICATION OF L VALUE ADDED TO ERROR MESSAGE.
      C       EVALUATION OF SIGNAL CHANNEL DENSITY BY NUMERICAL
      C       CONVOLUTION INTEGRAL OF THE BESSEL FUNCTIONS
      C
      C
            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
            DIMENSION WORK(100)
            EXTERNAL GRAND
            COMMON /INDEX/ L, LLL, LP1
            COMMON /PARMS/ GAMMA, DELTA, OODMO, MARY, FM, LL, FLL,
           $EHNO, EHNT, DELTAL, OODMOL, DMO
            RESULT=0.D0
      C USE ROMBERG INTEGRATION UNTIL CONTRIBUTIONS ARE SMALL
            XL=0.D0
            XU=1.D0
            DX=1.D0
      1     CALL DQATR(XL,XU,1.D-6,100,GRAND,PART,KODE,WORK)
            IF(KODE.EQ.0)GOTO 2
            IF(KODE.NE.1)THEN
              WRITE(6,100)L
              STOP 9999
      100     FORMAT(' DQATR ACCURACY NOT ATTAINED FOR L=',I3)
            ELSE
      C SUBDIVIDE THE INTERVAL
              XLS=XL
              XINC=(XU-XL)/8.D0
              PART=0.D0
              DO 200 ISI=1,8
              XUS=XLS+XINC
              IF(XUS.GT.XU)XUS=XU
              CALL DQATR(XLS,XUS,1.D-6,100,GRAND,PIECE,KODE,WORK)
              IF(KODE.EQ.0)GOTO 201
              WRITE(6,202)XLS,XUS,L,KODE
      202     FORMAT(' XLS=',1PD15.8,' XUS=',1PD15.8,'  L=',I2,
           $  ' DQATR ERROR KODE = ',I2)
              IF(KODE.NE.1)STOP 201
      201     PART=PART+PIECE
              XLS=XUS
      200     CONTINUE
            END IF
      2     RESULT=RESULT+PART
```

```
          IF(DABS(PART).LE.1.D-7*DABS(RESULT))GOTO 999
          XL=XU
          IF(DABS(PART/RESULT).LE.1.D-1)DX=DX*2.D0
          XU=XL+DX
          GOTO 1
   999    CONTINUE
          WRITE(6,199)XU
   199    FORMAT(' FINAL UPPER LIMIT = ',1PD24.16)
  7000    RETURN
          END
```



```
          DOUBLE PRECISION FUNCTION GRAND(X)
C
C PROBABILITY DENSITY FUNCTION FOR M-1 NOISE-ONLY CHANNELS
C
C THIS IS INTEGRAND FOR FINDING PROBABILITY THAT THE M-1
C NO-SIGNAL CHANNELS DO NOT EXCEED THE SIGNAL CHANNEL, GIVEN
C THE LEVEL AT THE DETECTOR OUTPUT OF THE SIGNAL CHANNEL
C
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          LOGICAL*1 ZLLLN, ZLM
          DIMENSION WORK2(100)
          EXTERNAL DENGND
          COMMON /PARMS/ GAMMA, DELTA, OODMO, MARY, FM, LL, FLL,
     $EHNO, EHNT, DELTAL, OODMOL, DMO
          COMMON /INDEX/ L, LLL, LP1
          COMMON /INDEX2/ M, N
          COMMON /POWERS/ XM, XN, EXPM, EXPN
          COMMON /BGP/ ZED
          COMMON /BINCS/ BLL1L1, BLL1L
          DX=DELTA*X
C ARGUMENT OF (MARY-1) POWER
          CALL DOSUM1(X,SUM1)
          CALL DOSUM2(X,SUM2)
          BASE=SUM1+SUM2
          POWM1=1.D0-DXI(BASE,MARY-1)
C
C DO THE DENSITY OF THE SIGNAL CHANNEL BY NUMERICAL INTEGRATION
C OF THE CONVOLUTION OF THE DENSITY FUNCTION FOR L HOPS JAMMED
C WITH THE DENSITY FUNCTION FOR (LL-L) HOPS UNJAMMED, TAKING
C INTO ACCOUNT THE DEGENERATE CASES FOR ALL HOPS JAMMED OR
C UNJAMMED, IN WHICH CASE A SIMPLE NONCENTRAL CHI-SQUARE DENSITY
C IS NEEDED WITHOUT THE BOTHER OF A CONVOLUTION
C
          ZED=X
          IF(L.EQ.0) THEN
C ... EVERYTHING IS UNJAMMED ...
             SCD=DELTA*CHI2NC(LL+LL,EHNO,DELTA*ZED)
          ELSE IF(L.EQ.LL) THEN
C ... EVERYTHING IS JAMMED ...
             SCD=CHI2NC(LL+LL,EHNT,ZED)
          ELSE
C ... SOME ARE JAMMED AND SOME ARE UNJAMMED, SO...
C            ...A CONVOLUTION IS NEEDED...
             NTERM=X
             IF(NTERM.LT.X.OR.NTERM.EQ.0)NTERM=NTERM+1
             DENINT=0.D0
             XL=0.D0
             DO 20001 NDEX=1,NTERM
             XU=NDEX
             IF(XU.GT.X)XU=X
```

```
                IF(NDEX.NE.1)THEN
                  CALL DQATR2(XL,XU,1.D-7,100,DENGND,DIPART,KODE,WORK2)
                  IF(KODE.EQ.0)GOTO 888
                  WRITE(5,887)KODE,XL,XU
        887       FORMAT(' DQATR2 KODE=',I2,' XL=',1PD10.3,' XU=',1PD10.3)
                  IF(KODE.NE.1)STOP 887
        888       DENINT=DENINT+DIPART
                ELSE
C SUBDIVIDE FIRST INTERVAL
                  XINC=(XU-XL)/5.DO
                  XLS=XL
                  DO 20002 MDEX=1,5
                  XUS=XLS+XINC
                  IF(XUS.GT.X)XUS=X
                  CALL DQATR2(XLS,XUS,1.D-7,100,DENGND,DIPART,KODE,WORK2)
                  IF(KODE.EQ.0)GOTO 889
                  IF(KODE.NE.1) THEN
                     WRITE(5,887)KODE,XLS,XUS
                     STOP 889
                  ELSE
C WE HAVE ROUND-OFF PROBLEMS, SO TRY RESUBDIVIDING INTERVAL...
                    XLA=XLS
                    XPLUS=(XUS-XLS)/5.DO
                    PIECE=0.DO
                    DO 20003 LDEX=1,5
                    XUA=XLA+XPLUS
                    IF(XUA.GT.X)XUA=X
                    CALL DQATR2(XLA,XUA,1.D-7,100,DENGND,CHUNK,KODE,WORK2)
                    IF(KODE.EQ.0)GOTO 870
                    WRITE(5,887)KODE,XLA,XUA
                    IF(KODE.NE.1)STOP 870
        870         PIECE=PIECE+CHUNK
                    XLA=XUA
        20003       CONTINUE
                    DIPART=PIECE
                  END IF
        889       DENINT=DENINT+DIPART
                  XLS=XUS
        20002     CONTINUE
                END IF
                XL=XU
        20001   CONTINUE
                XARGM=(FLL-L)*EHNO+L*EHNT+X
                XARGM=-XARGM
                SCD=DENINT*DEXP(XARGM)*DELTA
              END IF
              GRAND=SCD*POWM1
              RETURN
              END
```



```
              DOUBLE PRECISION FUNCTION GP(I,X)
C
C INCOMPLETE GAMMA FUNCTION P(A,X) FOR INTEGER VALUES OF A,
C EVALUATED USING P(A,X) = X**A x GAMMA*(A,X) WHERE GAMMA* IS THE
C INCOMPLETE GAMMA FUNCTION DEFINED IN EQ. 6.5.4 OF
C ABRAMOWITZ & STEGUN, HANDBOOK OF MATHEMATICAL FUNCTIONS,
C                      ----------------------------------
C WASHINGTON: GOVERNMENT PRINTING OFFICE, 1964.
C
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              DIMENSION FACTRL(33)
              DATA FACTRL / 1.0000000000000000D+00 ,
     $        2.0000000000000000D+00 , 6.0000000000000000D+00 ,
     $        2.4000000000000000D+01 , 1.2000000000000000D+02 ,
     $        7.2000000000000000D+02 , 5.0400000000000000D+03 ,
     $        4.0320000000000000D+04 , 3.6288000000000000D+05 ,
     $        3.6288000000000000D+06 , 3.9916800000000000D+07 ,
     $        4.7900160000000000D+08 , 6.2270208000000000D+09 ,
     $        8.7178291200000000D+10 , 1.3076743680000000D+12 ,
     $        2.0922789888000000D+13 , 3.5568742809600000D+14 ,
     $        6.4023737057280000D+15 , 1.2164510040883200D+17 ,
     $        2.4329020081766400D+18 , 5.1090942171709440D+19 ,
     $        1.1240007277776077D+21 , 2.5852016738884977D+22 ,
     $        6.2044840173323943D+23 , 1.5511210043330986D+25 ,
     $        4.0329146112660564D+26 , 1.0888869450418350D+28 ,
     $        3.0488834461171381D+29 , 8.8417619937397006D+30 ,
     $        2.6525285981219101D+32 , 8.2228386541779211D+33 ,
     $        2.6313083693369347D+35 , 8.6833176188118849D+36 /
              IF(X.EQ.0.D0)GOTO 900
C TEST FOR OVERFLOW REGION
              IF(X+I.GE.88.42D0)GOTO 1
C ...NO, JUST DO IT
              IF(I)1300,1200,1100
        1100  IF(I.GE.33)GOTO 1400
              F=FACTRL(I)
        1115  T=1.D0/F
              SUM=T
              N=0
        1120  N=N+1
              T=T*X/(I+N)
              SUM=SUM+T
              IF(T.LE.SUM*1.D-14)GOTO 1130
              GOTO 1120
        1130  SUM=SUM*DEXP(-X)
              GOTO 1140
        1200  SUM=1.D0
        1140  GIAST=SUM
        1210  GP=(X**I)*GIAST
              GOTO 999
C IF X=0 THEN RESULT IS 0
        900   GP=0.D0
        999   RETURN
```

```
C
C OVERFLOW REGION
C
C ...IS IT NEAR ENOUGH 1.DO TO APPROXIMATE IT BY THAT?
1       DLNX=DLOG(X)
        ARGMNT=X-(I-1)*DLNX
        IF(ARGMNT.GT.87.5DO)GOTO 2
C ...NO, USE ASYMPTOTIC FORMULA
        PART=DEXP(-ARGMNT)
        CHUNK=1.DO
        IF(I.LE.2)GOTO 3
        IM1=I-1
        PIECE=1.DO
        DO 4 J=1,IM1
        PIECE=PIECE*((I-J)/X)
        CHUNK=CHUNK+PIECE
        PART=PART/J
4       CONTINUE
3       GP=1.DO-PART*CHUNK
        GOTO 999
2       GP=1.DO
        GOTO 999
1300    WRITE(6,1001)
1001    FORMAT(' GIAST CALLED WITH NEG. INTEGER PARAMETER.')
        STOP 1300
1400    WRITE(6,1002)
1002    FORMAT(' GIAST I TOO LARGE, OVERFLOW.')
        STOP 1400
        END
```



```
        SUBROUTINE DOSUM1(X,SUM1)
C
C SUMMATION FOR DENSITY OF NOISE ONLY CHANNELS, R=1 TO L
C
        IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        COMMON /PARMS/ GAMMA, DELTA, OODMO, MARY, FM, LL, FLL,
     $EHNO, EHNT, DELTAL, OODMOL, DMO
        COMMON /INDEX/ L, LLL, LP1
        COMMON /BINCS/ BLL1L1, BLL1L
        COMMON /DELCON/ DC1, DC2, DC2I, DC3
C SUM NOT PRESENT IF L=0
        SUM1=0.DO
        IF(L.EQ.0)GOTO 999
C IF L=LL, ONLY LAST TERM IS NON-VANISHING
        IF(LLL)990,900,990
C ELSE MUST DO IT ALL
990     PARTBC=DMO*BLL1L1
        IF(MOD(L-1,2).NE.0)PART=-PART
        PVAL=GP(1,X)
        SUM1=PARTBC*PVAL
        IF(L.EQ.1)GOTO 800
C ELSE DO REST OF TERMS
        RECURS=DEXP(-X)
        DO 700 R=2,L
        FR=R
        PARTBC=PARTBC*((FR-LP1)/((FLL-FR)*OODMO))
        RECURS=RECURS*X/(FR-1.DO)
        PVAL=PVAL-RECURS
        TERM=PARTBC*PVAL
        SUM1=SUM1+TERM
700     CONTINUE
800     SUM1=SUM1*DC1
        GOTO 999
900     SUM1=GP(LL,X)
999     RETURN
        END
```

```
            SUBROUTINE DOSUM2(X,SUM2)
C
C SUMMATION FOR NOISE ONLY CHANNEL DENSITY, R=1 TO LL-L
C
            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
            COMMON /PARMS/ GAMMA, DELTA, OODMO, MARY, FM, LL, FLL,
        $EHNO, EHNT, DELTAL, OODMOL, DMO
            COMMON /INDEX/ L, LLL, LP1
            COMMON /BINCS/ BLL1L1, BLL1L
            COMMON /DELCON/ DC1, DC2, DC2I, DC3
C SUM NOT PRESENT IF LL-L=0
            SUM2=0.DO
            IF(LL.EQ.L)GOTO 999
C IF L=0, ONLY LAST TERM IS NON-VANISHING
            IF(L.EQ.0)GOTO 900
C ELSE MUST DO IT ALL
            BC=BLL1L
            FAC=DC2I
            PART=DC2I
            DX=DELTA*X
            PVAL=GP(1,DX)
            SUM2=BC*PART*PVAL
            IF(LLL.EQ.1)GOTO 800
C ELSE DO REST OF TERMS
            RECURS=DEXP(-DX)
            FLLL1=LLL+1
            DO 700 R=2,LLL
            FR=R
            BC=BC*(FLLL1-FR)/(FLL-FR)
            PART=PART*FAC
            RECURS=RECURS*DX/(FR-1.DO)
            PVAL=PVAL-RECURS
            TERM=BC*(PART*PVAL)
            SUM2=SUM2+TERM
  700       CONTINUE
  800       SUM2=SUM2*DC1
            IF(MOD(L,2).NE.0)SUM2=-SUM2
            GOTO 999
  900       SUM2=GP(LL,DELTA*X)
  999       RETURN
            END
```



```
            DOUBLE PRECISION FUNCTION DENGND(ETA)
C
C INTEGRAND FUNCTION FOR NUMERICAL CONVOLUTION OF THE INDIVIDUAL
C HOP DENSITIES FOR THE SIGNAL CHANNEL
C
            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
            COMMON /PARMS/ GAMMA, DELTA, OODMO, MARY, FM, LL, FLL,
        $EHNO, EHNT, DELTAL, OODMOL, DMO
            COMMON /BGP/ ZED
            COMMON /INDEX/ L, LLL, LP1
            SBN=DSQRT(4.DO*LLL*EHNO*DELTA*ETA)
            SBT=DSQRT(4.DO*L*EHNT*(ZED-ETA))
            IF(SBN.GT.60.DO.AND.SBN.GT.LLL-1)GOTO 10
C
C COMPUTE EITHER IO(X) OR EXP(-X)*IO(X) DEPENDING ON SIZE OF X
C
            CALL DBESI(SBN,LLL-1,SRN,KODEN)
            XF1=0.DO
            GOTO 11
  10        CALL DXBESI(SBN,LLL-1,SRN,KODEN)
            XF1=SBN
  11        IF(KODEN.EQ.0)GOTO 100
            WRITE(5,1)KODEN,SBN
  1         FORMAT(' DBESI FIRST CALL KODEN=',I2,' ARGUMENT = ',1PD10.3)
            IF(KODEN.NE.3)STOP 1010
  100       IF(SBT.GT.60.DO.AND.SBT.GT.L)GOTO 20
            CALL DBESI(SBT,L-1,SRT,KODET)
            XF2=0.DO
            GOTO 21
  20        CALL DXBESI(SBT,L-1,SRT,KODET)
            XF2=SBT
  21        IF(KODET.EQ.0)GOTO 200
            WRITE(5,2)KODET,SBT
  2         FORMAT(' DBESI SECOND CALL KODET=',I2,' ARGUMENT = ',1PD10.3)
            IF(KODET.NE.3)STOP 1020
  200       DENGND=SRN*SRT*DEXP(XF1+XF2-DMO*ETA)
            BASEN=DELTA*ETA/(LLL*EHNO)
            BASET=(ZED-ETA)/(L*EHNT)
            DENGND=DENGND*DXF(BASEN,(LLL-1.DO)/2.DO)*DXF(BASET,(L-1)/2.DO)
            RETURN
            END
```

```
0001            DOUBLE PRECISION FUNCTION CHI2NC(KF,ALPHA,Z)
        C
        C *********************************************************************
        C *                          W A R N I N G                            *
        C *                          -------------                            *
        C * THIS SUBROUTINE IS NOT A GENERAL NON-CENTRAL CHI-SQUARE DENSITY   *
        C *                          ---                                      *
        C *             ...DEGREES OF FREEDOM KF MUST BE EVEN...              *
        C *********************************************************************
        C
        C Z IS REALLY Z/2 IN THE USUAL FORMULATION OF THE DENSITY, FOR SPEED.
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            FREE=KF
0004            BARG=DSQRT(2.D0*FREE*ALPHA*Z)
0005            IF(BARG.GT.60.D0)GOTO 3
0006            CALL DBESI(BARG,KF/2-1,BRES,KODE)
0007            XF=0.D0
0008            GOTO 4
0009    3       CALL DXBESI(BARG,KF/2-1,BRES,KODE)
0010            XF=BARG
0011    4       IF(KODE.NE.0)WRITE(6,1)KODE,BARG
0012    1       FORMAT(' FROM CHI2NC, DBESI KODE = ',I1,' ARGUMENT = ',1PD10.3)
0013            IF(KODE.EQ.0)GOTO 2
0014            IF(KODE.NE.3)STOP 9100
0015    2       ARG=2.D0*Z/(FREE*ALPHA)
0016            XARG=Z+ALPHA+ALPHA
0017            CHI2NC=DXF(ARG,(FREE-2.D0)/4.D0)*DEXP(XF-XARG)*BRES
0018            RETURN
0019            END
```

APPENDIX 2H

COMPUTER PROGRAM TO COMPUTE
BIT ERROR PROBABILITY FOR
SQUARE-LAW LINEAR COMBINING RECEIVER
IN THE PARTIAL-BAND NOISE-JAMMING CHANNEL
USING SPECIAL-CASE EQUATION FOR L=1 HOP/SYMBOL

The following pages contain a listing of the FORTRAN-77 program used to obtain numerical values of the probability of error for the square-law linear combining receiver in the presence of partial-band noise jamming in the special case of L=1.

For listings of the subprograms UNDB and DLOG2, see Appendix 2G, listing pages 7 and 8.

```
0001          PROGRAM MFSKFH
        C-----------------------------------------------------------------------
        C PROGRAM TO COMPUTE THE EXACT EQUATION FOR THE BIT ERROR PROBABILITY
        C FOR MFSK/FH WITH L=1 HOP/SYMBOL IN THE PRESENCE OF PARTIAL BAND
        C NOISE JAMMING AND THERMAL NOISE.
        C
        C THE COMPUTATIONS ARE DONE BY THE SPECIAL CASE EQUATION FOR L=1. THEN
        C               P(ERROR) = 1 - P(CORRECT)
        C ON A SYMBOL BASIS.  THE CONVERSION TO BIT ERROR PROBABILITY ASSUMES
        C ORTHOGONAL SIGNALS, IN WHICH CASE:
        C
        C       P(BIT ERROR) = (M/(2M-2)) * P(SYMBOL ERROR)
        C-----------------------------------------------------------------------
        C PROGRAMMER: ROBERT H. FRENCH
        C            15 NOVEMBER 1982, 30 NOV. 1982
        C
        C VERSION 5.1.0 - SPECIAL CASE L=1, GENERAL M
        C
0002          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003          BYTE REPLY,YES,NO
0004          COMMON /INPUTS/ MLIST(5), LLIST(10), DEBNOL(5), DEBNJL(11),
             $ GAMLST(31)
0005          COMMON /OUTPUT/ PB(11)
0006          COMMON /SIZE/ NM, NL, NO, NJ, NG
0007          COMMON /PARMS/ GAMMA, DELTA, OODMO, MARY, FM, LL, FLL,
             $EHNO, EHNT, DELTAL, OODMOL, DMO
0008          COMMON /DELCON/ DC1, DC2, DC2I, DC3
0009          DATA YES,NO/'Y','N'/
0010    1001  WRITE(5,1000)
0011    1000  FORMAT(' DO YOU WANT UNDERFLOW MESSAGES? ',$)
0012          READ(5,1002)REPLY
0013    1002  FORMAT(A1)
0014          IF(REPLY.NE.YES.AND.REPLY.NE.NO)GOTO 1001
0015          IF(REPLY.EQ.YES)CALL ERRSET(74,.TRUE.,.FALSE.,.FALSE.,.TRUE.,255)
        C READ INPUT FROM TI:
0016          CALL GET
        C START TIMING THINGS AFTER INTERACTIVE INPUT....
0017          RTIME=SECNDS(0.)
        C
        C LOOP ON M (ORDER OF ALPHABET)
        C
0018          DO 900 IM=1,NM
0019          MARY=MLIST(IM)
0020          FM=MARY
0021          FK=DLOG2(FM)
0022          W2B=FM/(2.DO*(FM-1.DO))
        C
        C LOOP ON HOPS/SYMBOL
        C
0023          DO 800 IL=1,NL
0024          LL=LLIST(IL)
0025          FLL=LL
```



```
        C
        C LOOP ON EB/NO
        C
0026          DO 700 IO=1,NO
0027          EBNO=UNDB(DEBNOL(IO))
0028          ESNO=EBNO*FK
0029          EHNO=ESNO/FLL
0030          CALL PUT1(IM,IL,IO)
        C
        C LOOP ON GAMMA
        C
0031          DO 600 IG=1,NG
0032          GAMMA=GAMLST(IG)
        C
        C LOOP ON EB/NJ
        C
0033          DO 500 IJ=1,NJ
0034          EBNJ=UNDB(DEBNJL(IJ))
0035          ESNJ=EBNJ*FK
0036          EHNJ=ESNJ/FLL
0037          EHNT=GAMMA/(GAMMA/EHNO + 1.DO/EHNJ)
0038          DELTA=1.DO+EHNO/(GAMMA*EHNJ)
0039          DMO=DELTA-1.DO
0040          DELTAL=DLOG(DELTA)
0041          OODMO=1.DO/(DELTA-1.DO)
0042          OODMOL=DLOG(OODMO)
0043          DC2=DELTA*OODMO
0044          DC2I=1.DO/DC2
0045          DC3=1.DO-DELTA
0046          CALL PRCORR(PWC)
0047          PB(IJ)=W2B*(1.DO-PWC)
0048    500   CONTINUE
0049          CALL PUT2(IG)
0050    600   CONTINUE
0051    700   CONTINUE
0052    800   CONTINUE
0053    900   CONTINUE
0054          RTIME=SECNDS(RTIME)
0055          WRITE(6,1)RTIME
0056    1     FORMAT(///' ELAPSED WALL-CLOCK TIME = ',F10.2,' SECONDS.')
0057          STOP 0  .
0058          END
```

```
0001            SUBROUTINE GET
          C SUBROUTINE TO READ INPUT PARAMETERS FROM TI: DEVICE INTERACTIVELY
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            DIMENSION GAMDFL(31), SGL(31)
0004            LOGICAL*1 EJECT
0005            BYTE REPLY,YES,NO,BLANK
0006            COMMON /EJECT1/ EJECT
0007            COMMON /INPUTS/ MLIST(5), LLIST(10), DEBNOL(5), DEBNJL(11),
              $ GAMLST(31)
0008            COMMON /SIZE/ NM, NL, NO, NJ, NG
0009            DATA YES, NO, BLANK /'Y', 'N', ' '/
0010            DATA GAMDFL/
              $ 1.D-3,1.5D-3,2.D-3,3.D-3,4.D-3,5.D-3,6.D-3,7.D-3,8.D-3,9.D-3,
              $ 1.D-2,1.5D-2,2.D-2,3.D-2,4.D-2,5.D-2,6.D-2,7.D-2,8.D-2,9.D-2,
              $ 1.D-1,1.5D-1,2.D-1,3.D-1,4.D-1,5.D-1,6.D-1,7.D-1,8.D-1,9.D-1,
              $ 1.D0 /
0011            DATA SGL/
              $ 1.D-6,1.5D-6,2.D-6,3.D-6,4.D-6,5.D-6,6.D-6,7.D-6,8.D-6,9.D-6,
              $ 1.D-5,1.5D-5,2.D-5,3.D-5,4.D-5,5.D-5,6.D-5,7.D-5,8.D-5,9.D-5,
              $ 1.D-4,1.5D-4,2.D-4,3.D-4,4.D-4,5.D-4,6.D-4,7.D-4,8.D-4,9.D-4,
              $ 1.D-3 /
0012            WRITE(5,1)
0013      1     FORMAT(' ENTER 0 FOR NUMBER OF VALUES TO TAKE DEFAULTS IN [ ]')
0014      100   WRITE(5,2)
0015      2     FORMAT(' HOW MANY VALUES OF M? [M=2 ONLY] ',$)
0016            READ(5,3,ERR=100)NM
0017      3     FORMAT(I2)
0018            IF(NM.LT.0.OR.NM.GT.5)GOTO 100
0019            IF(NM.EQ.0)GOTO 103
0020            DO 102 IN=1,NM
0021      101   WRITE(5,4)IN
0022      4     FORMAT(5X,'M(',I1,') = ',$)
0023            READ(5,5,ERR=101)MLIST(IN)
0024      5     FORMAT(I5)
0025            IF(MLIST(IN).LE.1)GOTO 101
0026      102   CONTINUE
0027            GOTO 104
          C DEFAULT IS BINARY ONLY
0028      103   NM=1
0029            MLIST(1)=2
0030      104   NL=1
0031            LLIST(1)=1
0032      108   WRITE(5,8)
0033      8     FORMAT(' HOW MANY VALUES OF EB/NO? [13.35 DB ONLY] ',$)
0034            READ(5,3,ERR=108)NO
0035            IF(NO.LT.0.OR.NO.GT.5)GOTO 108
0036            IF(NO.EQ.0)GOTO 111
0037            DO 110 IN=1,NO
0038      109   WRITE(5,9)IN
0039      9     FORMAT(5X,'EB/NO(',I1,') (DB) = ',$)
0040            READ(5,10,ERR=109)DEBNOL(IN)
0041      10    FORMAT(F5.2)
0042      110   CONTINUE
```



```
0043            GOTO 112
          C DEFAULT IS EB/NO = 13.35 DB ONLY
0044      111   NO=1
0045            DEBNOL(1)=13.35
0046      112   WRITE(5,11)
0047      11    FORMAT(' HOW MANY VALUES OF EB/NJ? [0(5)50 DB] ',$)
0048            READ(5,5,ERR=112)NJ
0049            IF(NJ.LT.0.OR.NJ.GT.11)GOTO 112
0050            IF(NJ.EQ.0)GOTO 115
0051            DO 114 IN=1,NJ
0052      113   WRITE(5,12)IN
0053      12    FORMAT(5X,'EB/NJ(',I2,') (DB) = ',$)
0054            READ(5,10,ERR=113)DEBNJL(IN)
0055      114   CONTINUE
0056            GOTO 117
          C DEFAULT IS 0(5)50 DB
0057      115   NJ=11
0058            DO 116 IN=1,NJ
0059            DEBNJL(IN)=5*(IN-1)
0060      116   CONTINUE
0061      117   WRITE(5,13)
0062      13    FORMAT(' HOW MANY VALUES OF GAMMA? [.001 TO 1 IN 31 STEPS] '
               $ /'                        [-1 TO DIVIDE ABOVE BY 1000]',$)
0063            READ(5,3,ERR=117)NG
0064            IF(NG.EQ.-1)GOTO 130
0065            IF(NG.LT.0.OR.NG.GT.31)GOTO 117
0066            IF(NG.EQ.0)GOTO 120
0067            DO 119 IN=1,NG
0068      118   WRITE(5,14)IN
0069      14    FORMAT(5X,'GAMMA(',I2,') = ',$)
0070            READ(5,15,ERR=118)GAMLST(IN)
0071      15    FORMAT(D10.3)
0072            IF(GAMLST(IN).LE.0.D0.OR.GAMLST(IN).GT.1.D0)GOTO 118
0073      119   CONTINUE
0074            GOTO 122
          C DEFAULT .001,.0015,.002(.001).009,.1,.15,.02(.01).09,.1,.15,.2(.1)1.0
0075      120   NG=31
0076            DO 121 IN=1,NG
0077            GAMLST(IN)=GAMDFL(IN)
0078      121   CONTINUE
0079            GOTO 122
0080      130   NG=31
0081            DO 131 IN=1,NG
0082            GAMLST(IN)=SGL(IN)
0083      131   CONTINUE
0084      122   WRITE(5,16)
0085      16    FORMAT(' SUPPRESS PAGE EJECTS ON PRINT-OUT? [Y] ',$)
0086            READ(5,17)REPLY
0087      17    FORMAT(A1)
0088            IF(REPLY.NE.YES.AND.REPLY.NE.NO.AND.REPLY.NE.BLANK)GOTO 122
0089            IF(REPLY.EQ.BLANK.OR.REPLY.EQ.YES)EJECT=.FALSE.
0090            IF(REPLY.EQ.NO)EJECT=.TRUE.
0091            RETURN
0092            END
```

```
PDP-11 FORTRAN-77 V4.0-1        10:07:27    27-Feb-84          Page 5

              SUBROUTINE PUT1(IM,IL,IO)
      C
      C SUBROUTINE TO OUTPUT PAGE HEADERS
      C
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              LOGICAL*1 EJECT
              COMMON /EJECT1/ EJECT
              COMMON /INPUTS/ MLIST(5), LLIST(10), DEBNOL(5), DEBNJL(11),
           $ GAMLST(31)
              COMMON /SIZE/ NM, NL, NO, NJ, NG
              IF(EJECT)WRITE(6,999)
      999     FORMAT('1')
              WRITE(6,1)MLIST(IM),LLIST(IL)
      1       FORMAT(39X,'EXACT ANALYSIS FOR ',I2,'-ARY FSK/FH WITH L = ',
           $ I2,' HOPS/SYMBOL'//)
              WRITE(6,2)DEBNOL(IO)
      2       FORMAT(1X,'EB/NO = ',F7.3,' DB'//)
              WRITE(6,3)(DEBNJL(IOUT),IOUT=1,NJ)
      3       FORMAT(11X,<NJ>(' EB/NJ (DB)')/9X,<NJ>(5X,F6.2)/12X,
           $ <11*NJ-1>('-')/4X,'GAMMA',2X,<NJ>(5X,'PB',4X))
              RETURN
              END
```

```
PDP-11 FORTRAN-77 V4.0-1        10:07:31    27-Feb-84          Page 6

              SUBROUTINE PUT2(IG)
      C
      C WRITE ONE LINE OF RESULTS
      C
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              COMMON /INPUTS/ MLIST(5), LLIST(10), DEBNOL(5), DEBNJL(11),
           $ GAMLST(31)
              COMMON /OUTPUT/ PB(11)
              COMMON /SIZE/ NM, NL, NO, NJ, NG
              WRITE(6,1)GAMLST(IG),(PB(IOUT),IOUT=1,NJ)
      1       FORMAT(1X,1PD10.3,1P<NJ>D11.3)
              RETURN
              END
```

```
PDP-11 FORTRAN-77 V4.0-1        10:07:34    27-Feb-84          Page 7

              SUBROUTINE PRCORR(PWC)
      C
      C SUBROUTINE TO COMPUTE PROBABILITY OF CORRECT SYMBOL DECISION
      C
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              COMMON /PARMS/ GAMMA, DELTA, OODMO, MARY, FM, LL, FLL,
           $EHNO, EHNT, DELTAL, OODMOL, DMO
              COMMON /INDEX/ L, LLL, LP1
              COMMON /POWERS/ XM, XN, EXPM, EXPN
              COMMON /BINCS/ BLL1L1, BLL1L
              COMMON /DELCON/  DC1, DC2, DC2I, DC3
              OMG=1.DO-GAMMA
      C DO THE J=0 TERM OF THE SUM, WHICH IS ALWAYS 1.0
              SUM=1.DO
              PART=1.DO
      C DO THE REST OF THE SUM FROM 1 TO M-1
              DO 100 JP1=2,MARY
              J=JP1-1
              PART=-(MARY-J)*(PART/J)
              OOOPJ=1.DO/(J+1.DO)
              AJOJP1=-J*OOOPJ
              SUM=SUM+PART*(OMG*DEXP(AJOJP1*EHNO)+
           $ GAMMA*DEXP(AJOJP1*EHNT))*OOOPJ
      100     CONTINUE
              PWC=SUM
              RETURN
              END
```

APPENDIX 2I

COMPUTER PROGRAM TO COMPUTE BIT ERROR PROBABILITY FOR SQUARE-LAW LINEAR COMBINING RECEIVER IN THE PARTIAL-BAND NOISE-JAMMING CHANNEL USING SPECIAL-CASE EQUATION FOR L=2 HOPS/SYMBOL

The following pages contain a listing of the FORTRAN-77 program used to obtain numerical values of the probability of error for the square-law linear combining receiver in the presence of partial-band noise jamming in the special case of L=2 hops/symbol.

For a listing of subprogram DBINCO, see Appendix 4F, listing page 8.

The output format of this program requires a printer capable of printing 158 characters across a 14-inch page (e.g., a 12-pitch printout). This must also be taken into account when the task is built on an RSX-11M system (i.e., specify MAXBUF=158 as a task-builder option).

```
PDP-11 FORTRAN-77 V4.0-1          10:07:55    27-Feb-84           Page 1

0001            PROGRAM L2SC
        C FH/MFSK IN PARTIAL-BAND NOISE JAMMING, SPECIAL CASE FOR L=2
        C
        C PROGRAMMER: R. H. FRENCH            FORMULA: L. E. MILLER
        C               7 DEC 83, 8 DEC 83               6 DEC 83
        C
        C VERSION 1.2.0
        C
        C  STOP COMPUTING IF DELTA<1.0028 DUE TO ROUND-OFF PROBLEMS
        C  *** SEE COMPILER BUG WORK-AROUND IN SUBROUTINE PERR ***
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            CHARACTER*10 PEA(11),STARS
0004            LOGICAL*1 FOOTN
0005            COMMON /SIZE/ NO, NJ, NG
0006            COMMON /INPUTS/ DEBNOL(5), DEBNJL(11), GAMLST(31), MM, K, W2B
0007            COMMON /ANSWER/ PEA
0008            COMMON /PARMS/ DELTA, ESNO, ESNT
0009            DATA STARS/' *********'/
0010            CALL ERRSET(74,.TRUE.,.FALSE.,.FALSE.,.TRUE.,15)
0011            OPEN(UNIT=6,RECORDSIZE=158,TYPE='NEW')
0012            CALL GET
0013            AK=K
0014            DO 900 IO=1,NO
0015            EBNO=10.DO**(DEBNOL(IO)/10.DO)
0016            ESNO=AK*EBNO
0017            CALL PUT1(DEBNOL(IO))
0018            FOOTN=.FALSE.
0019            DO 800 IG=1,NG
0020            GAMMA = GAMLST(IG)
0021            DO 700 IJ=1,NJ
0022            EBNJ=10.DO**(DEBNJL(IJ)/10.DO)
0023            ESNJ=AK*EBNJ
0024            ESNT=GAMMA*ESNO*ESNJ/(ESNO+GAMMA*ESNJ)
0025            DELTA=ESNO/ESNT
0026            IF(DELTA.GT.1.0028DO.OR.GAMMA.EQ.1.DO) THEN
0027              CALL PERR(GAMMA,PROB)
0028              ENCODE(10,699,PEA(IJ),ERR=1000)PROB
0029    699       FORMAT(1PD10.3)
0030            ELSE
0031              FOOTN=.TRUE.
0032              PEA(IJ)=STARS
0033            END IF
0034    700     CONTINUE
0035            CALL PUT2(GAMMA)
0036    800     CONTINUE
0037            IF(FOOTN)WRITE(6,801)
0038    801     FORMAT(///' ********* = CAN NOT COMPUTE THIS POINT DUE TO',
             $ ' ROUND-OFF LIMITATIONS')
0039    900     CONTINUE
0040            STOP 0
0041    1000    WRITE(5,1001)
0042    1001    FORMAT(' I/O ERROR WHILE ENCODING P(E) FOR OUTPUT')
0043            STOP 1
0044            END
```

```
PDP-11 FORTRAN-77 V4.0-1          10:08:01    27-Feb-84           Page 2

0001            SUBROUTINE PUT1(DEBNO)
        C
        C WRITE PAGE HEADERS
        C
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            COMMON /SIZE/ NO, NJ, NG
0004            COMMON /INPUTS/ DEBNOL(5), DEBNJL(11), GAMLST(31), MM, K, W2B
0005            MTITLE=15+(NJ*13-12)/2
0006            MLU=13*NJ-4
0007            WRITE(6,1) MM,DEBNO,(DEBNJL(I),I=1,NJ)
0008    1       FORMAT('1',38X,'EXACT PERFORMANCE OF ',I2,'-ARY FSK/FH WITH'
             $ ' L=2 HOPS/',
             $ 'SYMBOL (SPECIAL CASE EQUATION)'//1X,'EB/NO = ',F7.4,' dB'//
             $ <MTITLE>X,'P(E) FOR'/15X,<MLU>('-')/10X,<NJ>(6X,'EB/NJ =')/
             $ 4X,'GAMMA',2X,<NJ>(4X,F6.2,' dB')/2X,'---------',<NJ>(4X,
             $ '---------'))
0009            RETURN
0010            END
```

```
PDP-11 FORTRAN-77 V4.0-1          10:08:04    27-Feb-84           Page 3

0001            SUBROUTINE PUT2(GAMMA)
        C
        C WRITE LINE OF RESULTS
        C
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            CHARACTER*10 PEA(11)
0004            COMMON /ANSWER/ PEA
0005            COMMON /SIZE/ NO, NJ, NG
0006            WRITE(6,1)GAMMA,(PEA(I),I=1,NJ)
0007    1       FORMAT(1X,1PD10.3,<NJ>(3X,A10))
0008            RETURN
0009            END
```

```
0001          SUBROUTINE GET
        C
        C INTERACTIVE INPUT OF PARAMETERS
        C
0002          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003          CHARACTER*1 BLANK,VA15(15)
0004          CHARACTER*8 VALUE, BLANK8
0005          CHARACTER*15 VAL15, BLNK15
0006          DIMENSION DNL(3,5), DGAM(31)
0007          EQUIVALENCE(VAL15,VA15(1))
0008          COMMON /SIZE/ NO, NJ, NG
0009          COMMON /INPUTS/ DEBNOL(5), DEBNJL(11), GAMLST(31), MM, K, W2B
        C DEFAULT EB/NO (DB) VALUES FOR (ROWS) M=2,4,8,16,32 AND (COLUMNS)
        C     P(E|L=1) =    1.D-3        1.D-4        1.D-5
0010          DATA DNL/ 10.9444D0, 12.3133D0, 13.3525D0,
       $                 8.3524D0,  9.6284D0, 10.6065D0,
       $                 6.9718D0,  8.1690D0,  9.0939D0,
       $                 6.0696D0,  7.1996D0,  8.0783D0,
       $                 5.4183D0,  6.4910D0,  7.3295D0 /
0011          DATA DGAM/
       $ 1.D-3,1.5D-3,2.D-3,3.D-3,4.D-3,5.D-3,6.D-3,7.D-3,8.D-3,9.D-3,
       $ 1.D-2,1.5D-2,2.D-2,3.D-2,4.D-2,5.D-2,6.D-2,7.D-2,8.D-2,9.D-2,
       $ 1.D-1,1.5D-1,2.D-1,3.D-1,4.D-1,5.D-1,6.D-1,7.D-1,8.D-1,9.D-1,
       $ 1.D0 /
0012          DATA BLANK/' '/
0013          DATA BLANK8/'        '/, BLNK15/'               '/
0014    1     WRITE(5,2)
0015    2     FORMAT(' ENTER NUMBER OF BITS/SYMBOL (K) [2]: ',$)
0016          READ(5,3,ERR=1)K
0017    3     FORMAT(I1)
0018          IF(K.EQ.0)K=2
0019          IF(K.LT.0)GOTO 1
0020          MM=2**K
0021          W2B=0.5D0*MM/(MM-1.D0)
0022    4     WRITE(5,5)
0023    5     FORMAT(' ENTER NUMBER OF VALUES OF EB/NO [3]: ',$)
0024          READ(5,6,ERR=4)NO
0025    6     FORMAT(I1)
0026          IF(NO.EQ.0)NO=3
0027          IF(NO.LT.0.OR.NO.GT.5)GOTO 4
0028          DO 11 IN=1,NO
0029          IF(K.LE.5.AND.IN.LE.3) THEN
0030            DB = DNL(IN,K)
0031          ELSE
0032            DB = 0.D0
0033          END IF
0034    7     WRITE(5,8)IN,DB
0035    8     FORMAT('   ENTER EB/NO(',I1,') [',F7.4,' dB]: ',$)
0036          READ(5,9,ERR=7)VALUE
0037    9     FORMAT(A8)
0038          IF(VALUE.EQ.BLANK8) THEN
0039            DEBNOL(IN)=DB
```



```
0040          ELSE
0041            DECODE(8,10,VALUE,ERR=7)DEBNOL(IN)
0042    10      FORMAT(F8.4)
0043          END IF
0044    11    CONTINUE
0045    12    WRITE(5,13)
0046    13    FORMAT(' ENTER NUMBER OF VALUES OF EB/NJ [11]: ',$)
0047          READ(5,14,ERR=12)NJ
0048    14    FORMAT(I1)
0049          IF(NJ.EQ.0)NJ=11
0050          IF(NJ.LT.0.OR.NO.GT.11)GOTO 12
0051          DO 19 IN=1,NJ
0052          DB=(IN-1)*5
0053    15    WRITE(5,16)IN,DB
0054    16    FORMAT('   ENTER EB/NJ(',I2,') [',F5.2,' dB]: ',$)
0055          READ(5,17,ERR=15)VALUE
0056    17    FORMAT(A8)
0057          IF(VALUE.EQ.BLANK8) THEN
0058            DEBNJL(IN)=DB
0059          ELSE
0060            DECODE(8,18,VALUE,ERR=15)DEBNJL(IN)
0061    18      FORMAT(F8.4)
0062          END IF
0063    19    CONTINUE
0064    20    WRITE(5,21)
0065    21    FORMAT(' ENTER NUMBER OF VALUES OF GAMMA [31]: ',$)
0066          READ(5,22,ERR=20)NG
0067    22    FORMAT(I2)
0068          IF(NG.EQ.0)NG=31
0069          IF(NG.LT.0.OR.NG.GT.31)GOTO 20
0070          DO 27 IN=1,NG
0071    23    WRITE(5,24)IN,DGAM(IN)
0072    24    FORMAT('   ENTER GAMMA(',I2,') [',1PD8.1,']: ',$)
0073          READ(5,25,ERR=23)VAL15
0074    25    FORMAT(A15)
0075          IF(VAL15.EQ.BLNK15) THEN
0076            GAMLST(IN)=DGAM(IN)
0077          ELSE
        C THIS KLUGE IS DUE TO DEC FORTRAN-77 NOT SUPPORTING THE
        C CONCATENATION OPERATION ON CHARACTER VARIABLES...
0078            DO 251 JUST=1,15
0079            IF(VA15(15).NE.BLANK) GOTO 252
0080            DO 250 MOVE=1,14
0081            VA15(16-MOVE)=VA15(15-MOVE)
0082    250     CONTINUE
0083            VA15(1)=BLANK
0084    251     CONTINUE
0085    252     DECODE(15,26,VAL15,ERR=23)GAMLST(IN)
0086    26      FORMAT(D15.8)
0087            IF(GAMLST(IN).LE.0.D0.OR.GAMLST(IN).GT.1.D0)GOTO 23
0088          END IF
0089    27    CONTINUE
0090          RETURN
0091          END
```

```
        DOUBLE PRECISION FUNCTION DGL(X,K,N)
C
C DOUBLE PRECISION LAGUERRE POLYNOMIALS
C
        IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        NPA=N+K
        PART=DBINCO(NPA,N)
        DGL=PART
        IF(N.EQ.0)RETURN
        FNP1=N+1
        AK=K
        DO 100 M=1,N
        FM=M
        PART=-PART*(X/FM)*((FNP1-FM)/(AK+FM))
        DGL=DGL+PART
100     CONTINUE
        RETURN
        END
```



```
        SUBROUTINE PERR(GAMMA,PE)
C
C SUBROUTINE TO COMPUTE PROBABILITY OF ERROR
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        COMMON /INPUTS/ DEBNOL(5), DEBNJL(11), GAMLST(31), MM, K, W2B
        COMMON /PARMS/ DELTA, ESNO, ESNT
        OMG=1.DO-GAMMA
        OMG2=OMG*OMG
        G2=GAMMA*GAMMA
        TGOMG=2.DO*GAMMA*OMG
        AKM1=DELTA-1.DO
        AK=DELTA
        AKOK=DELTA/AKM1
        FMM=MM
        MM1=MM-1
C SUM OF ALL-JAMMED AND ALL-UNJAMMED PROBABILITIES
        PARTM=-1.DO
        SUM1=0.DO
        DO 900 M=1,MM1
        FM=M
        FM1=FM+1.DO
        PARTM=-((FMM-FM)/FM)*PARTM
        ARGLO=-ESNO/FM1
        ARGLT=-ESNT/FM1
        FMOM1=FM/FM1
        XO=OMG2*DEXP(-FMOM1*ESNO)
        XT=G2*DEXP(-FMOM1*ESNT)
        SUMP=0.DO
        DO 800 IP=0,M
        P=IP
        IF(IP.EQ.0) THEN
          PARTP=1.DO
          PIECE1=1.DO/(FM1*FM1)
        ELSE
          PARTP=PARTP*((FM1-P)/P)
C *****************************************************************
C *** THE DUMMY VARIABLE IN THE NEXT TWO STATEMENTS IS REQUIRED ***
C *** TO WORK AROUND A COMPILER BUG IN THE DEC FORTRAN-77 V4.0  ***
C *****************************************************************
          DUMMY=PIECE1*(P/FM1)
          PIECE1=DUMMY
        END IF
        GER1=DGL(ARGLO,1,IP)
        GER2=DGL(ARGLT,1,IP)
        ENDS=PIECE1*(XO*GER1+XT*GER2)
        TERMP=PARTP*ENDS
        SUMP=SUMP+TERMP
800     CONTINUE
        TERMM=SUMP*PARTM
        SUM1=SUM1+TERMM
900     CONTINUE
```

```
C SUM OF 1-JAMMED, 1-UNJAMMED TERMS
          SUM2=0.D0
          DO 700 M=0,MM1
          FM=M
          FMMM=MM-M
          IF(M.EQ.0) THEN
            PARTM=-1.D0
            LOWP=1
          ELSE
            PARTM=-((FMM-FM)/FM)*PARTM
            LOWP=0
          END IF
          LASTP=MM1-M
          SUMP=0.D0
          DO 600 IP=LOWP,LASTP
          P=IP
          IF(IP.EQ.LOWP) THEN
            PARTP=DBINCO(MM1-M,IP)*(AKOK**M)*DELTA*TGOMG/AKM1**IP
          ELSE
            PARTP=PARTP*((FMMM-P)/P)/AKM1
          END IF
          ANUM=DELTA*P+FM
          DENOM1=ANUM+DELTA
          DENOM2=ANUM+1.D0
          COEF=PARTP/(DENOM1*DENOM2)
          CXA=0.5D0*ESNO*ANUM/DENOM1+0.5D0*ESNT*ANUM/DENOM2
          TERMP=COEF*DEXP(-CXA)
          SUMP=SUMP+TERMP
  600     CONTINUE
          TERMM=SUMP*PARTM
          SUM2=SUM2+TERMM
  700     CONTINUE
          SUM=SUM1+SUM2
          PE=W2B*SUM
          RETURN
          END
```

APPENDIX 4A

COEFFICIENTS OF A POWER SERIES RAISED TO A POWER

In order to solve an integral for the error probability it was necessary to evaluate the power series expansion of

$$\left[e_{L-1}(x)\right]^m = \left[\sum_{n=0}^{L-1} \frac{x^n}{n!}\right]^m . \tag{4A-1}$$

A very useful formula, due to J.C.P. Miller, was found in [3, p. 42]:

$$(1 + b_1 x + b_2 x^2 + \ldots)^m = \sum_{r=0}^{\infty} d_r \, x^r \tag{4A-2}$$

where

$$d_r = \frac{1}{r} \sum_{n=1}^{r} [(m+1)n-r] \, d_{r-n} \, b_n . \tag{4A-3}$$

For the case of (4A-1) the coefficients b_n are

$$b_n = \begin{cases} 1/n!, & n=0, 1, \ldots, L-1 \\ 0, & n > L-1. \end{cases} \tag{4A-4}$$

Thus $$d_r = \frac{1}{r} \sum_{n=1}^{\min(r,L-1)} [(m+1)n-r] \, \frac{d_{r-n}}{n!} \, ; \quad d_0 = 1. \tag{4A-5}$$

We may use a slightly different definition of the coefficients to write

$$\left[e_{L-1}(x)\right]^m = \sum_{r=0}^{m(L-1)} \frac{c_r(m,L)}{r!} \, x^r . \tag{4A-6}$$

The coefficients c_r are simply

$$c_r(m,L) = r!\, d_r$$

$$= \frac{r!}{r} \sum_{n=1}^{\min(r,L-1)} [(m+1)n-r] \cdot \frac{c_{r-n}(m,L)}{(r-n)!} \cdot \frac{1}{n!}$$

$$= \frac{1}{r} \sum_{n=1}^{\min(r,L-1)} \binom{r}{n} [(m+1)n-r]\, c_{r-n}(m,L); \; c_0 = 1. \qquad (4A\text{-}7)$$

It is not difficult to show that

$$c_r(m,L) = m^r, \quad r \leq L-1; \qquad (4A\text{-}8)$$

and that

$$c_r(m,2) = \frac{m!}{(m-r)!} \,. \qquad (4A\text{-}9)$$

APPENDIX 4B

THE EDGEWORTH SERIES

The complexity in finding the probability density function (pdf) and cumulative distribution function (cdf) of the decision variables z_i, $i = 1, 2, \ldots, M$, is mainly due to the complexity of performing L-fold convolutions. When the random variables (RVs) to be convolved are chi-squared distributed, the resulting probability distribution is merely another chi-squared. Thus, by its unique property, the problem becomes much simplified. However, when the RVs to be convolved are either Rician or Rayleigh distributed, a closed form expression for the resulting pdf or cdf is not available. This appendix gives asymptotic expansions with respect to L for the pdf and cdf of

$$x = \frac{y - \bar{y}}{\sqrt{\mathrm{Var}(y)}} = \frac{\sum_{k=1}^{L}\left(y_k - \bar{y}_k\right)}{\sum_{k=1}^{L}\left[\mathrm{Var}\left(y_k\right)\right]^{1/2}} \tag{4B-1}$$

where y_k, $k = 1, 2, \ldots, L$, are L independent RVs with means $\bar{y}_k$, variances $\mathrm{var}(y_k)$, and higher order cumulants $\kappa_{r,k}$.

The Gram-Charlier series for the pdf of x is given by

$$p_x(\alpha) = \sum_{n=0}^{\infty} a_n\, Z(\alpha)\, H_n(\alpha), \quad -\infty < x < \infty. \tag{4B-2}$$

where the

$$H_n(\alpha) = (-1)^n \frac{d^n}{d\alpha^n}[Z(\alpha)]/Z(\alpha) \tag{4B-3}$$

are the Hermite polynomials* of degree n, and

*A more common definition of the Hermite polynomials $H_n(\alpha)$ is equivalent to $2^{n/2} H_n(\alpha\sqrt{2})$ in terms of (4B-3); in these cases the polynomials defined by (4B-3) are sometimes designated [4, eq. 22.5.18], [21, p. 189] as $He_n(\alpha)$ or $\mathcal{H}_n(\alpha)$.

$$Z(\alpha) = \frac{1}{\sqrt{2\pi}} e^{-\alpha^2/2}, \; -\infty < \alpha < \infty \tag{4B-4}$$

is the normalized Gaussian pdf.

The coefficients a_n can be obtained by multiplying both sides of (4B-2) by $H_n(\alpha)$ and integrating from $-\infty$ to ∞. Thus,

$$a_n = \frac{1}{n!} \int_{-\infty}^{\infty} p_x(\alpha)\, H_n(\alpha)\, d\alpha. \tag{4B-5}$$

Using the power series expansion of $H_n(\alpha)$ given by

$$H_n(\alpha) = \alpha^n - \frac{n(n-1)}{2 \cdot 1!} \alpha^{n-2} + \frac{n(n-1)(n-2)(n-3)}{2^2 \cdot 2!} \alpha^{n-4}$$

$$- \frac{n(n-1)(n-2)(n-3)(n-4)(n-5)}{2^3 \cdot 3!} \alpha^{n-6} + \ldots \tag{4B-6}$$

and substituting into (4B-5), we have

$$a_n = \frac{1}{n!} \left[\mu_n' - \frac{n(n-1)}{2 \cdot 1!} \mu_{n-2}' + \frac{n(n-1)(n-2)(n-3)}{2^2 \cdot 2} \mu_{n-4}' - \ldots \right] \tag{4B-7}$$

where μ_n' is the nth moment of $p_x(\alpha)$ about the origin. Normalizing $p_x(\alpha)$ to conform with (4B-1) to zero mean and unit variance, we have

$$a_0 = 1 \tag{4B-8a}$$

$$a_1 = 0 \tag{4B-8b}$$

$$a_2 = 0 \tag{4B-8c}$$

$$a_3 = \frac{1}{6} \mu_3 \tag{4B-8d}$$

$$a_4 = \frac{1}{24} (\mu_4 - 6\mu_2 + 3) = \frac{1}{24} (\mu_4 - 3) \tag{4B-8e}$$

where μ_n is the nth moment of $p_x(\alpha)$ about the mean. The pdf of x, then, is

$$p_x(\alpha) = Z(\alpha) - \left[\frac{\gamma_1}{6} Z^{(3)}(\alpha)\right] + \left[\frac{\gamma_2}{24} Z^{(4)}(\alpha) + \frac{\gamma_1^2}{72} Z^{(6)}(\alpha)\right]$$

$$- \left[\frac{\gamma_3}{120} Z^{(5)}(\alpha) + \frac{\gamma_1\gamma_2}{144} Z^{(7)}(\alpha) + \frac{\gamma_1^3}{1296} Z^{(9)}(\alpha)\right]$$

$$+ \left[\frac{\gamma_4}{720} Z^{(6)}(\alpha) + \frac{\gamma_2^2}{1152} Z^{(8)}(\alpha) + \frac{\gamma_1\gamma_3}{720} Z^{(8)}(\alpha) + \frac{\gamma_1^2\gamma_2}{1728} Z^{(10)}(\alpha)\right.$$

$$\left. + \frac{\gamma_1^4}{31104} Z^{(12)}(\alpha)\right] + \ldots \quad . \tag{4B-9}$$

This version of the Gram-Charlier series, in which the brackets enclose terms with equal order of magnitude with respect to L, is the Edgeworth series. The cdf F(x) may be obtained by integrating (4B-9). Thus

$$F_x(\alpha) = P(\alpha) - \left[\frac{\gamma_1}{6} Z^{(2)}(\alpha)\right] + \left[\frac{\gamma_2}{24} Z^{(3)}(\alpha) + \frac{\gamma_1^2}{72} Z^{(5)}(\alpha)\right]$$

$$- \left[\frac{\gamma_3}{120} Z^{(4)}(\alpha) + \frac{\gamma_1\gamma_2}{144} Z^{(6)}(\alpha) + \frac{\gamma_1^3}{1296} Z^{(8)}(\alpha)\right]$$

$$+ \left[\frac{\gamma_4}{720} Z^{(5)}(\alpha) + \frac{\gamma_2^2}{1152} Z^{(7)}(\alpha) + \frac{\gamma_1\gamma_3}{720} Z^{(7)}(\alpha) + \frac{\gamma_1^2\gamma_2}{1728} Z^{(9)}(\alpha)\right.$$

$$\left. + \frac{\gamma_1^4}{31104} Z^{(11)}(\alpha)\right] + \ldots, \tag{4B-10}$$

where

$$P(x) = \int_{-\infty}^{x} Z(t)\, dt \tag{4B-11}$$

is the Gaussian cumulative distribution function.

The coefficients* γ_n may be expressed in terms of higher order cumulants of y_k, $k = 1, 2, \ldots, L$. Thus

$$\gamma_{r-2} = \frac{1}{L^{r/2-1}} \frac{\frac{1}{L}\sum_{k=1}^{L} \kappa_{r,k}}{\left[\frac{1}{L}\sum_{k=1}^{L} \mathrm{Var}(y_k)\right]^{\frac{1}{2}}}, \tag{4B-12}$$

with cumulants (for the kth hop) as follows:

$$\kappa_1 = E\{y_k\} \tag{4B-13a}$$

$$\kappa_2 = E\{y_k^2\} - \kappa_1^2 \tag{4B-13b}$$

$$\kappa_3 = E\{y_k^3\} - 3\kappa_2\kappa_1 - \kappa_1^3 \tag{4B-13c}$$

$$\kappa_4 = E\{y_k^4\} - 4\kappa_3\kappa_1 - 3\kappa_2^2 - 6\kappa_2\kappa_1^2 - \kappa_1^4 \tag{4B-13d}$$

$$\kappa_5 = E\{y_k^5\} - 5\kappa_4\kappa_1 - 10\kappa_2\kappa_3 - 10\kappa_3\kappa_1^2 - 15\kappa_1\kappa_2^2 - 10\kappa_\sigma\kappa_1^3 - \kappa_1^5 \tag{4B-13e}$$

$$\kappa_6 = E\{y_k^5\} - 6\kappa_5\kappa_1 - 15\kappa_2\kappa_4 - 10\kappa_3^2 - 15\kappa_4\kappa_1^2 - 60\kappa_1\kappa_2\kappa_3$$
$$- 15\kappa_2^3 - 20\kappa_3\kappa_1^3 - 45\kappa_1^2\kappa_2^2 - 15\kappa_2\kappa_1^4 - \kappa_1^6. \tag{4B-13f}$$

Equations (4B-9) and (4B-10) thus express the pdf and cdf of the RV x in (4B-1) in terms of the moments anu cumulants of the samples y_k, $k = 1, \ldots, L$, by the pdf of Gaussian RV and its derivatives in an asymptotic series expansion.

*The use of the symbol γ_n for these coefficients follows conventional usage, e.g. [4, p. 935]; there is no implied connection between these coefficients and the partial-band fraction γ as used in the main text.

AD-A147 766 OPTIMUM JAMMING EFFECTS ON FREQUENCY-HOPPING M-ARY FSK 6/7
(FREQUENCY-SHIFT K..(U) LEE (J S) ASSOCIATES INC
ARLINGTON VA J S LEE ET AL. OCT 84 JC-2025-N
UNCLASSIFIED N00014-83-C-0312 F/G 17/4 NL

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

APPENDIX 4C

APPLICATIONS OF THE EDGEWORTH SERIES TO EVALUATION OF THE PROBABILITY OF ERROR FOR THE LINEAR-LAW AGC RECEIVER

In this Appendix we explain the numerical method used in obtaining the probability of error for the linear-law AGC receiver using the Edgeworth series.

The equation to be computed is

$$P_b(e) = \frac{M/2}{(M-1)} \sum_{\ell=0}^{L} \binom{L}{\ell} \gamma^{\ell} (1-\gamma)^{L-\ell}$$

$$\cdot \left\{ 1 - \int_0^{\infty} \frac{1}{\sqrt{\sum_{k=1}^{L} \text{Var } z_{1k}}} \, p_x\left(\frac{\alpha - \sum_{k=1}^{L} \overline{z_{1k}}}{\sqrt{\sum_{k=1}^{L} \text{Var } z_{1k}}} \right) \right.$$

$$\left. \cdot \left[F_x\left(\frac{\alpha - \sum_{k=1}^{L} \overline{z_{2k}}}{\sqrt{\sum_{k=1}^{L} \text{Var } z_{2k}}} \right) \right]^{M-1} d\alpha \right\} \qquad \text{(4C-1)}$$

where $p_x(\alpha)$ and $F_x(\beta)$ are given in (4B-9) and (4B-10) of Appendix 4B. For the signal channel,

$$x = \frac{\sum_{k=1}^{L} \left(z_{1k} - \overline{z_{1k}} \right)}{\sum_{k=1}^{L} \left[\text{Var}\left(z_{1k} \right) \right]^{1/2}} . \qquad \text{(4C-2)}$$

The nth moment of z_{1k} is

$$E\left\{z_{1k}^{n}\right\} = 2^{n/2}\,\Gamma\left(\frac{n}{2}+1\right)\,{}_1F_1\left(-\frac{n}{2};\,1;\,-\rho_k/L\right) \tag{4C-3}$$

where

$$\rho_k = \begin{cases} \rho_N & \text{for unjammed cell with probability } (1-\gamma) \\ \rho_T & \text{for jammed cell with probability } \gamma. \end{cases} \tag{4C-4}$$

For the noise-only channels,

$$x = \frac{\sum_{k=1}^{L}\left(z_{2k} - \overline{z_{2k}}\right)}{\left[\sum_{k=1}^{L} \mathrm{Var}\left(z_{2k}\right)\right]^{\frac{1}{2}}} \tag{4C-5}$$

The nth moment of this normalized Rayleigh distributed RV z_{2k} is

$$E\left\{z_{2k}^{n}\right\} = \begin{cases} \sqrt{\frac{\pi}{2}}\; 1\cdot 3\cdot \ldots \cdot n, & \text{for } n \text{ odd} \\ 2^{p}\; p!\,, & \text{for } n = 2p. \end{cases} \tag{4C-6}$$

Since the moments of z_{2k} are independent of σ_{2k} or ρ, the jammed case and the unjammed case have the same contribution to the error probability from the noise channels. This is due to the normalization used by the AGC receiver.

Having obtained the moments of z_{1k} and z_{2k} for both jammed and unjammed situations, the procedure to obtain the probability of bit error expression can be easily traced by the self explanatory flow-chart given in Figure 4C-1.

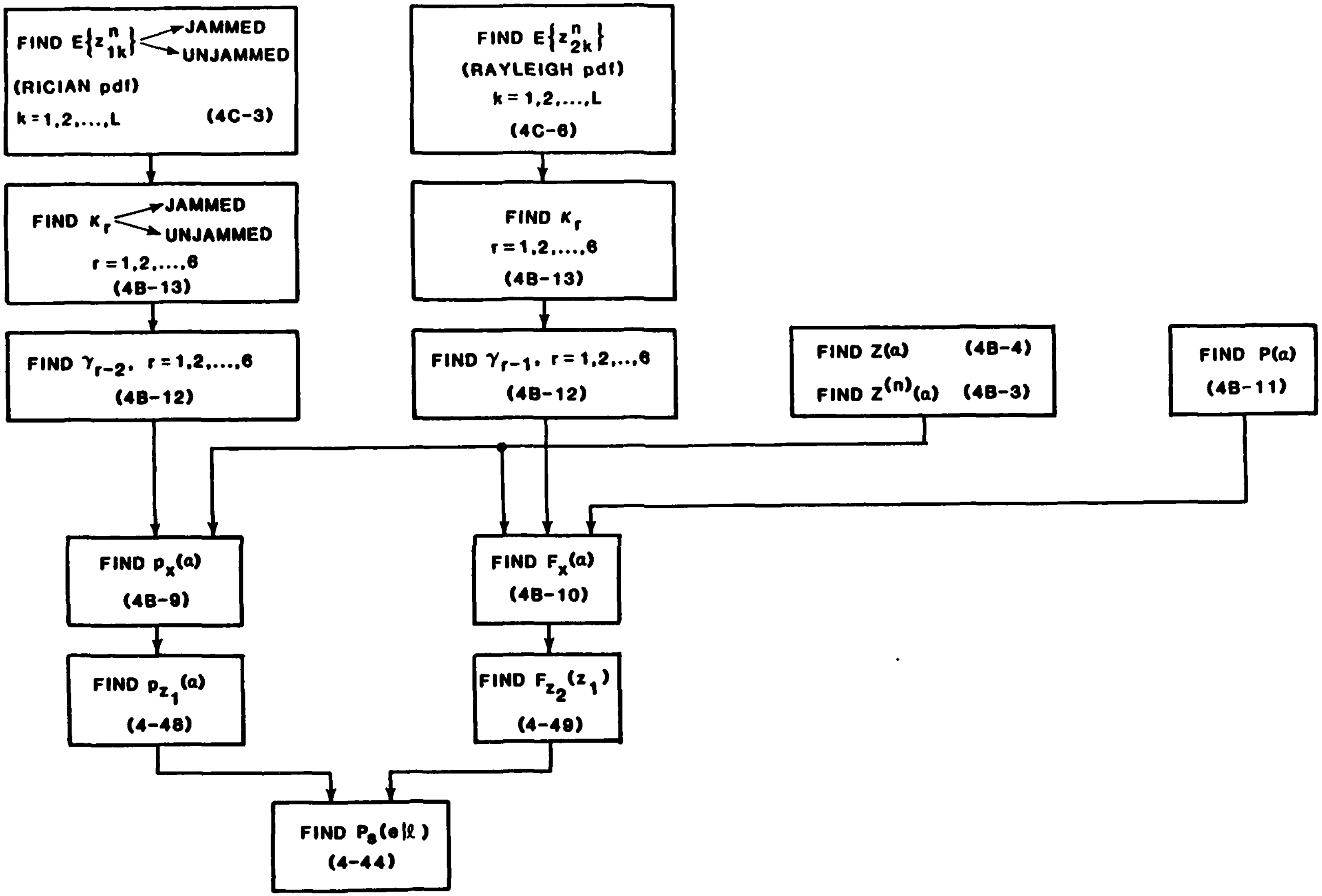


FIGURE 4C-1 COMPUTATIONAL FLOWCHART

APPENDIX 4D

SPECIAL CASES OF LINEAR-LAW AGC RECEIVER PERFORMANCE EQUATION FOR L=1 AND L=2

When L=1, the decision variables are merely

$$z_i = z_{i1}; \ i = 1, 2, \ldots, M. \tag{4D-1}$$

The probability of error expression is then

$$P_b(e) = \frac{M/2}{(M-1)} \left\{ (1-\gamma) \left\{ 1 - \int_0^\infty p_{z_{11}}\left(\alpha, \rho_N\right) \left[\int_0^\alpha p_{z_{21}}(\beta) d\beta \right]^{M-1} d\alpha \right\} \right.$$

$$\left. + \gamma \left\{ 1 - \int_0^\infty p_{z_{11}}\left(\alpha, \rho_T\right) \left[\int_0^\alpha p_{z_{21}}(\beta) d\beta \right]^{M-1} d\alpha \right\} \right\} \tag{4D-2}$$

which is readily integrated numerically using the computer program given by the listing in Appendix 4F.

When L=2, the decision variables are

$$z_i = z_{i1} + z_{i2}; \ \ i = 1, 2, \ldots, M. \tag{4D-3}$$

Since we assume that the z_{i1} and the z_{i2} are independent, the pdf of z_i is given by $p_{z_{i1}} \circledast p_{z_{i2}}$ where $\circledast$ denotes the convolution operation.

The convolution of the normalized Rayleigh pdf's is

$$p_{z_2}(\alpha) = \int_0^\alpha x(\alpha - x)\, e^{-x^2/2}\, e^{-(\alpha - x)^2/2}\, dx. \tag{4D-4}$$

With algebraic manipulations and the change of variable $u = x - \alpha/2$,

$$p_{z_2}(\alpha) = 2e^{-\alpha^2/4} \int_0^{\alpha/2} \left(\frac{\alpha^2}{4} - u^2 \right) \exp(-u^2) du, \tag{4D-5}$$

which can be expressed in terms of the error function [2, eq. 3.321.2 and 3.381.1], [4, eq. 6.5.16 and 6.5.22] as

$$p_{z_2}(\alpha) = e^{-\alpha^2/4} \sqrt{\pi} \; \mathrm{erf}(\alpha/2) \left(\frac{\alpha^2}{4} - \frac{1}{2}\right) + (\alpha/2)e^{-\alpha^2/2}, \; \alpha > 0. \tag{4D-6}$$

The resulting error probability can then be numerically computed by the formulation

$$P_b(e) = \frac{M/2}{M-1} \sum_{\ell=0}^{2} \binom{2}{\ell} \gamma^{\ell}(1-\gamma)^{2-\ell}$$

$$\cdot \left\{1 - \int_0^{\infty} p_{z_{11}} \circledast p_{z_{12}} \, d\alpha_1 \left[\int_0^{\alpha_1} p_{z_2}(\beta) \, d\beta\right]^{M-1}\right\} \tag{4D-7}$$

in which $p_{z_{1k}}(\alpha)$ is given by (4-47a) with

$$\left.\begin{aligned} &\rho_1 = \rho_2 = S/\sigma_N^2 \text{ for } \ell=0 \\ &\rho_1 = S/\sigma_N^2, \quad \rho_2 = S/\sigma_T^2 \text{ for } \ell=1 \\ &\text{and} \\ &\rho_1 = \rho_2 = S/\sigma_N^2 \text{ for } \ell=2. \end{aligned}\right\} \tag{4D-8}$$

APPENDIX 4E

COMPUTER PROGRAM TO COMPUTE BIT ERROR PROBABILITY FOR SQUARE-LAW FH/MFSK RECEIVER WITH AGC

The following pages contain a listing of the FORTRAN-77 program used to calculate numerical values for the error probability of a square-law FH/MFSK receiver with AGC in the presence of partial-band and wideband noise jamming.

The default increments of E_b/N_J in dB are chosen to facilitate plotting on a scale of 7 divisions = 5 dB.

```
      PROGRAM MAGEX
C COMPUTE AGC RECEIVER PERFORMANCE IN THE PRESENCE OF OPTIMUM
C PARTIAL-BAND JAMMING AND FULL-BAND JAMMING FOR MFSK/FH
C
C PROGRAMMERS:  L.E.MILLER AND R.H.FRENCH
C               9 FEBRUARY 1983
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL PJ
      COMMON /PARMS/ RHON, EBNO, EBNJ, FLL, LL, FMM, MM, FKK
      COMMON /SIZE/ NO, NJ, NL
      COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(5)
      COMMON /ANSWER/ PE(5), GAMMA(5)
      CALL GET
      DO 900 IO=1,NO
      EBNO=10.D0**(DEBNO(IO)/10.D0)
      CALL PUT1(IO)
      DO 800 IJ=1,NJ
      EBNJ=10.D0**(DEBNJ(IJ)/10.D0)
      DO 600 IL=1,NL
      LL=LLIST(IL)
      FLL=LL
      RHON=EBNO/FLL
      PE(IL)=PJ(1.D0)
      GAMMA(IL)=1.D0
600   CONTINUE
      CALL PUT2(IJ)
      DO 700 IL=1,NL
      LL=LLIST(IL)
      FLL=LL
      RHON=EBNO/FLL
      CALL MAX01(PJ,PJMAX,GMAX,0.1D0,0.5D0)
      PE(IL)=PJMAX
      GAMMA(IL)=GMAX
700   CONTINUE
      CALL PUT2(IJ)
800   CONTINUE
900   CONTINUE
      STOP 0
      END
```



```
      SUBROUTINE GET
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION RLIST(5)
      COMMON /PARMS/ RHON, EBNO, EBNJ, FLL, LL, FMM, MM, FKK
      COMMON /SIZE/ NO, NJ, NL
      DATA RLIST/13.3525,10.6065,9.0939,8.0783,7.3295/
      COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(5)
20    WRITE(5,21)
21    FORMAT(' WHAT VALUE OF K? [1]', $)
      READ(5,3,ERR=20)KK
      IF(KK.LT.0.OR.KK.GT.5)GOTO 20
      IF(KK.EQ.0) KK=1
      FKK=KK
      MM=2**KK
      RATIO=RLIST(KK)
      FMM=MM
1     WRITE(5,2)RATIO
2     FORMAT(' HOW MANY VALUES OF EB/NO? [',F7.4,' DB ONLY] ',$)
      READ(5,3,ERR=1)NO
3     FORMAT(I2)
      IF(NO.LT.0.OR.NO.GT.10)GOTO 1
      IF(NO.NE.0) THEN
      DO 7 IN=1,NO
4     WRITE(5,5)IN
5     FORMAT('  EB/NO(',I2,') (DB) = ',$)
      READ(5,6,ERR=4)DEBNO(IN)
6     FORMAT(F7.2)
7     CONTINUE
      ELSE
C DEFAULT IS RATIO DB ONLY
      NO=1
      DEBNO(1)=RATIO
      END IF
8     WRITE(5,9)
9     FORMAT(' HOW MANY VALUES OF EB/NJ? [0(25/7)50 DB] ',$)
      READ(5,3,ERR=8)NJ
      IF(NJ.LT.0.OR.NJ.GT.50)GOTO 8
      IF(NJ.NE.0)THEN
      DO 12 IN=1,NJ
10    WRITE(5,11)IN
11    FORMAT('  EB/NJ(',I2,') (DB) = ',$)
      READ(5,6,ERR=10)DEBNJ(IN)
12    CONTINUE
      ELSE
C SET UP DEFAULT LIST
      NJ=15
      DO 13 IN=1,NJ
      DEBNJ(IN)=25.*(IN-1)/7.
13    CONTINUE
      END IF
14    WRITE(5,15)
15    FORMAT(' HOW MANY VALUES OF L? [1,2,3,4,6] ',$)
```

PDP-11 FORTRAN-77 V4.0-1   13:36:11   14-May-84   Page 3

```
          READ(5,3,ERR=14)NL
          IF(NL.LT.0.OR.NL.GT.5)GOTO 14
          IF(NL.NE.0)THEN
          DO 19 IN=1,NL
16        WRITE(5,17)IN
17        FORMAT('  L(',I2,') = ',$)
          READ(5,18,ERR=16)LLIST(IN)
18        FORMAT(I5)
19        CONTINUE
          ELSE
          NL=5
          LLIST(1)=1
          LLIST(2)=2
          LLIST(3)=3
          LLIST(4)=4
          LLIST(5)=6
          END IF
          RETURN
          END
```

PDP-11 FORTRAN-77 V4.0-1   13:36:19   14-May-84   Page 4

```
          SUBROUTINE PUT1(IO)
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          COMMON /PARMS/ RHON, EBNO, EBNJ, FLL, LL, FMM, MM, FKK
          COMMON /SIZE/ NO, NJ, NL
          COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(5)
          WRITE(6,1)DEBNO(IO), MM
1         FORMAT('1',39X,
     $ 'EXACT PERFORMANCE OF AGC RECEIVER(M-ARY CASE)'/2X,
     $ 'EB/NO = ',F6.2,' dB',10X,'M = ',I2)
          WRITE(6,2)(LLIST(K),K=1,NL)
2         FORMAT(//6X,<NL>(10X,'L = ',I2,8X)/1X,'EB/NJ',<NL>(2X,
     $ 22('-'))/1X,'(dB)',1X,<NL>(5X,'P(E)',8X,'GAMMA',2X)/1X,
     $ '-----',<NL>(2X,'----------',2X,'----------'))
          RETURN
          END
```

PDP-11 FORTRAN-77 V4.0-1   13:36:21   14-May-84   Page 5

```
          SUBROUTINE PUT2(IJ)
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          COMMON /SIZE/ NO, NJ, NL
          COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(5)
          COMMON /ANSWER/ PE(5), GAMMA(5)
          WRITE(6,1)DEBNJ(IJ),(PE(K),GAMMA(K),K=1,NL)
1         FORMAT(1X,F5.2,1P10D12.3)
          RETURN
          END
```

PDP-11 FORTRAN-77 V4.0-1   13:36:23   14-May-84   Page 6

```
          DOUBLE PRECISION FUNCTION PJ(GAMMA)
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          COMMON /PARMS/ RHON, EBNO, EBNJ, FLL, LL, FMM,MM, FKK
          RHOT=FKK*GAMMA*EBNO*EBNJ/(FLL*(EBNO+GAMMA*EBNJ))
          OMG=1.DO-GAMMA
C IF 1-GAMMA = 0 THEN ONLY L=LL TERM IS NONVANISHING
          IF(OMG.NE.0.DO)GOTO 50
          RHOL=FLL*RHOT
          SUM=CP(RHOL)
          GOTO 200
C SUM FOR L=0 TERM
50        PART=OMG**LL
          RHOL=FLL*FKK*RHON
          SUM=PART*CP(RHOL)
C BYPASS SUM OF ZERO TERMS IF GAMMA = 0
          IF(GAMMA.EQ.0)GOTO 200
          DO 100 L=1,LL
          FL=L
          PART=PART*((FLL-FL+1.DO)/FL)*GAMMA/OMG
          RHOL=FL*RHOT+(FLL-FL)*FKK*RHON
          TERM=PART*CP(RHOL)
          SUM=SUM+TERM
100       CONTINUE
200       PJ=SUM
          RETURN
          END
```

```
      DOUBLE PRECISION FUNCTION CP(RHOL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION FLG(10),COEF2(10)
      COMMON /PARMS/ RHON, EBNO, EBNJ, FLL, LL, FMM, MM, FKK
      MMM=MM-1
      FFM=MMM
      COEF1=-1.DO
      SUM1=0.
      DO 400 M=1,MMM
      FM=M
      COEF1=-COEF1*(FMM-FM)/FM
      TERM1=COEF1*EXP(-RHOL*FM/(1.DO+FM))/(1.DO+FM)**LL
      SUM2=1.DO
      IF(LL.EQ.1)GOTO 400
      FLG(1)=1.DO
      ARG=RHOL/(1.DO+FM)
      FLG(2)=FLL+ARG
      COEF3=1./(1.DO+FM)
      COEF2(9)=FM*COEF3
      COEF2(8)=1.DO
      SUM2=1.DO+COEF2(9)*FLG(2)
      MR=M*(LL-1)
      IF(MR.LT.2) GOTO 350
      DO 350 IR=2,MR
      FIR=IR
      FLG(3)=(2.DO*FIR+FLL-2.DO+ARG)*FLG(2)/FIR
      FLG(3)=FLG(3)-(FIR+FLL-2.DO)*FLG(1)/FIR
      FLG(1)=FLG(2)
      FLG(2)=FLG(3)
      JRR=IR
      IF(IR.GE.LL) JRR=LL-1
      COEF4=1./FIR
      SUM3=0.
      DO 300 JR=1,JRR
      FJR=JR
      COEF4=COEF4*COEF3*(FIR-FJR+1.DO)/FJR
      SUM3=SUM3+COEF4*((FM+1.DO)*FJR-FIR)*COEF2(10-JR)
300   CONTINUE
      COEF2(10)=SUM3
      SUM2=SUM2+COEF2(10)*FLG(3)
      DO 310 JR=1,9
310   COEF2(JR)=COEF2(JR+1)
350   CONTINUE
400   SUM1=SUM1+TERM1*SUM2
      CP=SUM1*FMM/2./FFM
      RETURN
      END
```



```
      SUBROUTINE MAX01(F,FMAX,XMAX,STEP,GUESS)
C THIS SEARCHES FOR MAX OF F(X) FOR 0 <= X <= 1
C   THIS ROUTINE WILL HAVE TROUBLE IF:
C        A) THE FUNCTION F(X) HAS MULTIPLE LOCAL MAXIMA IN [0,1]
C   OR   B) THE FUNCTION IS VERY STEEP AND THE INPUT 'STEP'
C            IS TOO LARGE
C   INPUT PARAMETERS:
C          F  NAME OF FUNCTION TO BE MAXIMIZED (EXTERNAL
C             DECLARATION REQUIRED IN CALLING PROGRAM)
C       FMAX  MAXIMUM VALUE OF FUNCTION F(X) OVER [0,1] (OUTPUT)
C       XMAX  THE VALUE OF X FOR WHICH F(XMAX) = FMAX   (OUTPUT)
C       STEP  INITIAL STEP SIZE FOR SEARCH              (INPUT)
C      GUESS  INITIAL GUESS AT XMAX, 0 <= GUESS <= 1    (INPUT)
C
C RESTRICTION: THE FUNCTION F MUST BE A DOUBLE PRECISION FUNCTION
C              OF A DOUBLE PRECISION ARGUMENT
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      X=GUESS
      DX=STEP
      TEST=1.D-4
      IF(DABS(STEP).LE.TEST)TEST=DABS(STEP)*1.D-4
      IF(TEST.EQ.0.DO)TEST=1.D-8
10    FO=F(X)
      IF(X.EQ.1.DO) THEN
        DX=-DABS(DX)
        F1=F(X+DX)
      ELSE IF(X.EQ.0.DO)THEN
        DX=DABS(DX)
        F1=F(X+DX)
      ELSE
        IF(X+DX.GT.1.DO) THEN
          DX=(1.DO-X)/10.DO
        ELSE IF(X-DX.LT.0.DO) THEN
          DX=X/10.DO
        ELSE
          CONTINUE
        END IF
        F1=F(X+DX)
C ARE WE GOING THE RIGHT WAY?
        IF(F1.GE.FO)GOTO 100
C ...NO, SWITCH DIRECTION
        DX=-DX
        F1=F(X+DX)
        IF(F1.GE.FO)GOTO 100
C ELSE WE MUST BE CLOSE TO A MAX AT X=GUESS OR TOO CLOSE
C TO AN ENDPOINT SO CUT STEP SIZE AND TRY AGAIN
        DXO=DX
        DX=DX/10.DO
C CLOSE ENOUGH, CALL IT A MAX AT X=GUESS
C UNLESS NOT TIGHT ENOUGH TEST...
        IF(DABS(X/DXO).LT.1000.DO)GOTO 11
```

```
12        XMAX=X
          FMAX=F0
          RETURN
C MUST TIGHTEN TEST CONSTANT
11        IF(DX.EQ.0.D0.OR.X.EQ.0.D0)GOTO 12
          TEST=TEST/10.
          GOTO 10
        END IF
C AVOID PROBLEMS AT ENDPOINTS
400     IF((X.EQ.0.D0.OR.X.EQ.1.D0).AND.F1.LE.F0)THEN
          DX=DX/10.D0
          IF(DABS(DX).LE.TEST) THEN
C MAX MUST BE END POINT
            FMAX=F0
            XMAX=X
            RETURN
          END IF
          F1=F(X+DX)
          GO TO 400
        END IF
C NOW GOING RIGHT DIRECTION.   KEEP GOING UNTIL PASS MAXIMUM.
100     X2=X+DX+DX
C HAVE WE REACHED END POINT?
        IF(X2.GT.1.D0.OR.X2.LT.0.D0)GOTO 110
C ELSE ALL OK
105     F2=F(X2)
C PAST MAX?
        IF(F2.LE.F1)GOTO 200
C ...NO, STEP AGAIN
        F0=F1
        F1=F2
        X=X+DX
        GOTO 100
C MAX MAY BE AT ENDPOINT.  CUT STEP SIZE AND
C TRY AGAIN IF NOT ALREADY TOO SMALL AN INCREMENT
110     IF(DABS(DX).LE.TEST)GOTO 120
        DX=DX/10.D0
        F0=F1
        GOTO 116
115     DX=DX/10.D0
116     F1=F(X+DX)
        GOTO 100
C MAX MUST BE AT ENDPOINT (OR WITHIN MINIMUM DX OF AN ENDPOINT)
120     IF(X2.LE.0.D0)GOTO 122
C MAX AT X=1.
        XMAX=1.D0
121     FMAX=F(XMAX)
        RETURN
C MAX AT X=0.
122     XMAX=0.D0
        GOTO 121
```



```
C HAVE PASSED MAX. IS IT LOCATED CLOSE ENOUGH YET?
200     IF(DABS(DX).LE.TEST)GOTO 300
C ...NO, CUT STEP SIZE AND TRY AGAIN
        GOTO 115
C DONE...
C SINCE  F0<=F1 & F2<=F1 AND ABS(DX) < MIN DX, CALL F1 THE MAX
300     IF(DABS(X+DX).LT.100.D0*DABS(TEST))GOTO 301
        FMAX=F1
        XMAX=X+DX
        RETURN
301     TEST=TEST/10.D0
        GOTO 115
        END
```

APPENDIX 4F

COMPUTER PROGRAM TO COMPUTE BIT ERROR PROBABILITY FOR LINEAR-LAW FH/MFSK RECEIVER WITH AGC USING EXACT EQUATION FOR L=1 HOP/SYMBOL

The following pages contain a listing of the FORTRAN-77 program used to calculate numerical values for the error probability of a linear-law FH/MFSK receiver with AGC in the presence of partial-band noise jamming for L=1 hop/symbol.

For subroutine DGAU, see Appendix 4G, listing page 11. For subroutines ATTACH and DETACH, see Appendix 4H, listing pages 20 and 21. A listing of function DXI is given in Appendix 4I. For subroutine DXBESI, see Appendix 4G, listing pages 12-13.

PDP-11 FORTRAN-77 V4.0-1 11:36:53 17-Aug-83 Page 1

```
              PROGRAM LINAGC
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                     C
      C THIS PROGRAM COMPUTES THE ERROR PERFORMANCE OF FH/MFSK C
      C    AGC RECEIVER EMPLOYING LINEAR ENVELOPE DETECTORS    C
      C        USING EXACT EQUATION FOR L=1 HOP/SYMBOL         C
      C                                                     C
      C PROGRAMMER: A. KADRICHU                               C
      C                                                     C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              LOGICAL*1 TERM
              EXTERNAL PEG
              COMMON /VARS/ RHOO, RHOJ
              COMMON /AK/ FKK
              COMMON /TOTAL/ RHOT
              COMMON /MARY/M,FM
              COMMON /HOPS/ LL, FLL
              COMMON /SIZE/ NO, NJ, NL
              COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(1)
              DIMENSION GAMDAT(28)
              DATA GAMDAT/  .001D0,.002D0,.003D0,.004D0,.005D0
          $               , .006D0,.007D0,.008D0,.009D0
          $               , .01D0 ,.02D0 ,.03D0 ,.04D0 ,.05D0
          $               , .06D0 ,.07D0 ,.08D0 ,.09D0
          $               , .1D0  ,.2D0  ,.3D0  ,.4D0  ,.5D0
          $               , .6D0  ,.7D0  ,.8D0  ,.9D0  ,1.D0/
              CALL ATTACH(6,TERM,IIIIII)
              CALL GET
              LL=LLIST(1)
              FLL=LL
              DO 800 IO=1,NO
              EBNO=10.D0**(DEBNO(IO)/10.D0)
              DO 700 IJ=1,NJ
              CALL PUT1(M)
              EBNJ=10.D0**(DEBNJ(IJ)/10.D0)
              RHOO=EBNO*FKK/FLL
              RHOJ=EBNJ*FKK/FLL
              CALL PUT2(LL,DEBNO(IO),DEBNJ(IJ))
              DO 1144 IG=1,16
              IGMA=29-IG
              GM=GAMDAT(IGMA)
              RHOT=GM*RHOO*RHOJ/(RHOO+GM*RHOJ)
              PERR=PEG(GM)
              WRITE(6,1)PERR,GM
      1       FORMAT(1X,'PE= ',1PD12.3, '  GM= ',D12.3)
      1144    CONTINUE
      700     CONTINUE
      800     CONTINUE
              CALL DETACH(6,TERM,IIIIII)
              STOP 0
              END
```

PDP-11 FORTRAN-77 V4.0-1 11:36:58 17-Aug-83 Page 2

```
              SUBROUTINE GET
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                C
      C INTERACTIVE INPUT OF PARAMETERS C
      C                                C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              COMMON /MARY/M,FM
              COMMON /AK/ FKK
              COMMON /SIZE/ NO, NJ
              DIMENSION RLIST(5)
              DATA RLIST/13.3525,10.6065,9.0939,8.0783,7.3295/
              COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(1)
      20      WRITE(5,21)
      21      FORMAT(' WHAT VALUE OF K? [1]', $)
              READ(5,3,ERR=20)KK
              IF(KK.LT.0.OR.KK.GT.5)GOTO 20
              IF(KK.EQ.0) KK=1
              FKK=KK
              M=2**KK
              RATIO=RLIST(KK)
              FM=M
      1       WRITE(5,2)RATIO
      2       FORMAT(' HOW MANY VALUES OF EB/NO? [',F7.4,' DB ONLY] ',$)
              READ(5,3,ERR=1)NO
      3       FORMAT(I2)
              IF(NO.LT.0.OR.NO.GT.10)GOTO 1
              IF(NO.NE.0) THEN
                DO 7 IN=1,NO
      4         WRITE(5,5)IN
      5         FORMAT('  EB/NO(',I2,') (DB) = ',$)
                READ(5,6,ERR=4)DEBNO(IN)
      6         FORMAT(F7.2)
      7         CONTINUE
              ELSE
      C DEFAULT IS RATIO DB ONLY
                NO=1
                DEBNO(1)=RATIO
              END IF
      8       WRITE(5,9)
      9       FORMAT(' HOW MANY VALUES OF EB/NJ? [0(25/7)50 DB] ',$)
              READ(5,3,ERR=8)NJ
              IF(NJ.LT.0.OR.NJ.GT.50)GOTO 8
              IF(NJ.NE.0)THEN
                DO 12 IN=1,NJ
      10        WRITE(5,11)IN
      11        FORMAT('  EB/NJ(',I2,') (DB) = ',$)
                READ(5,6,ERR=10)DEBNJ(IN)
      12        CONTINUE
```

```
PDP-11 FORTRAN-77 V4.0-1          11:36:58     17-Aug-83          Page 3

0044            ELSE
        C SET UP DEFAULT LIST
0045              NJ=15
0046              DO 13 IN=1,NJ
0047              DEBNJ(IN)=25.*(IN-1)/7.
0048      13      CONTINUE
0049            END IF
0050            LLIST(1)=1
0051            RETURN
0052            END
```

```
PDP-11 FORTRAN-77 V4.0-1          11:37:05     17-Aug-83          Page 4

0001            SUBROUTINE PUT1(M)
        CCCCCCCCCCCCCCCCCCCCCC
        C                    C
        C WRITE PAGE HEADERS C
        C                    C
        CCCCCCCCCCCCCCCCCCCCCC
0002            WRITE(6,1)M
0003      1     FORMAT('1'//I3,' MFSK/FH WITH AGC USING EXACT EQUATION')
0004            RETURN
0005            END
```

```
PDP-11 FORTRAN-77 V4.0-1          11:37:07     17-Aug-83          Page 5

0001            SUBROUTINE PUT2(LL,DEBNO,DEBNJ)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                            C
        C WRITE OUT INPUT PARAMETERS C
        C                            C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            WRITE(6,1)LL,DEBNO,DEBNJ
0004      1     FORMAT(' L = ',I2,' HOPS/BIT',5X,'EB/NO = ',F6.2,' DB',5X,
              $ 'EB/NJ = ',F6.2,' DB')
0005            RETURN
0006            END
```

```
PDP-11 FORTRAN-77 V4.0-1          11:37:08     17-Aug-83          Page 6

0001            DOUBLE PRECISION FUNCTION PEG(GM)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                              C
        C SUBROUTINE TO CALCULATE P(E) AS FUNCTION OF C
        C      SINGLE ARGUMENT GM WITH INTEGRAND       C
        C        FUNCTION DEFINED IN GRAND(Z1)         C
        C                                              C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            EXTERNAL GRAND
0004            DIMENSION WORK(100)
0005            COMMON /HOPS/ LL, FLL
0006            COMMON /MARY/M,FM
0007            COMMON /VARS/ RHOO, RHOJ
0008            COMMON /NEW/ RHO
0009            COMMON /TOTAL/ RHOT
0010          . SUMC=0.DO
0011            DO 1000 IL=0,LL
0012            IF( IL.EQ.0) THEN
0013              RHO=RHOO
0014            ELSE IF(IL.EQ.1) THEN
0015              RHO=RHOT
0016            ENDIF
0017            SUM=0.DO
0018            XL=0.DO
0019      100   XU=XL+1.DO
0020            CALL DGAU(XL,XU,GRAND,RESULT)
0021              SUM=SUM+RESULT
0022              IF(DABS(RESULT).LE.1.D-5*DABS(SUM))GOTO 200
0023            XL=XU
0024            GOTO 100
0025      200   PE=SUM
0026            YL=DBINCO(LL,IL)
0027            SUMC=SUMC+YL*GM**IL*DXI(1.DO-GM,LL-IL)*PE
0028      1000  CONTINUE
0029            PEG=1.DO-SUMC
0030            PEG=FM*PEG*.5DO/(FM-1.DO)
0031            WRITE(5,1040)GM,PEG
0032      1040  FORMAT(' GAMMA = ',1PE15.8,'   PEG = ',E15.8)
0033            RETURN
0034            END
```

PDP-11 FORTRAN-77 V4.0-1   11:37:13   17-Aug-83   Page 7

```
              DOUBLE PRECISION FUNCTION GRAND(Z1)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                            C
C PRODUCT OF THE PDF OF RANDOM VARIABLE {Z1} C
C      WITH THE Pr{Zi<Z1}, i=2,...,M          C
C             Z1---SIGNAL CHANNEL             C
C             Zi---NOISE CHANNELS             C
C                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              COMMON /MARY/ M,FM
              COMMON /NEW/ RHO
              S=DSQRT(2.D0*RHO)
              A=1.D0-DEXP(-Z1*Z1/2.D0)
              B=-RHO-Z1*Z1/2.D0+Z1*S
              S=DSQRT(2.D0*RHO)
              CALL DXBESI(Z1*S,0,ANS1,KODE)
              GRAND=Z1*(A**(M-1))*DEXP(B)*ANS1
              RETURN
              END
```

PDP-11 FORTRAN-77 V4.0-1   11:37:15   17-Aug-83   Page 8

```
              DOUBLE PRECISION FUNCTION DBINCO(N,K)
CCCCCCCCCCCCCCCCCCCCCCCCCCC
C                         C
C  BINOMIAL COEFFICIENTS  C
C                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              IF(K.GT.N) GOTO 6
              IF(K.EQ.N .OR. K.EQ.0)  GOTO1
              IF(K.EQ.1 .OR. K.EQ.N-1) GOTO2
              IF(K.GT.N/2) GOTO3
              KK=K
      5       C=N+1
              A=N
              DO 4 J=2,KK
              A=A*(C-J)/J
      4       CONTINUE
              DBINCO=A
              GOTO7
      3       KK=N-K
              GOTO 5
      2       DBINCO=N
              GOTO 7
      1       DBINCO=1.D0
              GOTO 7
      6       DBINCO=0.D0
      7       RETURN
              END
```

APPENDIX 4G

COMPUTER PROGRAM TO COMPUTE BIT ERROR PROBABILITY FOR LINEAR-LAW FH/MFSK RECEIVER WITH AGC USING EXACT EQUATION FOR L=2 HOPS/SYMBOL

The following pages contain a listing of the FORTRAN-77 program used to calculate numerical values for the error probability of a linear-law FH/MFSK receiver with AGC in the presence of partial-band noise jamming for L=2 hops/symbol.

For subroutines ATTACH and DETACH, see Appendix 4H, listing pages 20 and 21. A listing of function DXI is given in Appendix 4I. Subroutine DGAU20 is a second copy of DGAU (supplied from the system library) under a different name to avoid recursion in performing the double integration.

The constant PIR in function GR2 is $\sqrt{\pi}$.

```
            PROGRAM LINAGC
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                         C
      C THIS PROGRAM COMPUTES THE ERROR PERFORMANCE OF FH/MFSK  C
      C    AGC RECEIVER EMPLOYING LINEAR ENVELOPE DETECTORS     C
      C        USING EXACT EQUATION FOR L=2 HOPS/SYMBOL         C
      C                                                         C
      C PROGRAMMER: A. KADRICHU                                  C
      C                                                         C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            LOGICAL*1 TERM
            EXTERNAL PEG
            COMMON /VARS/ RHOO, RHOJ
            COMMON /AK/ FKK
            COMMON /TOTAL/ RHOT
            COMMON /MARY/M,FM
            COMMON /HOPS/ LL, FLL
            COMMON /SIZE/ NO, NJ
            COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(1)
            DIMENSION GAMDAT(28)
            DATA GAMDAT/  .001D0,.002D0,.003D0,.004D0,.005D0
     $                  , .006D0,.007D0,.008D0,.009D0
     $                  , .01D0 ,.02D0 ,.03D0 ,.04D0 ,.05D0
     $                  , .06D0 ,.07D0 ,.08D0 ,.09D0
     $                  , .1D0  ,.2D0  ,.3D0  ,.4D0  ,.5D0
     $                  , .6D0  ,.7D0  ,.8D0  ,.9D0  ,1.D0/
            CALL ATTACH(6,TERM,IIIIII)
            CALL GET
            LL=LLIST(1)
            FLL=LL
            DO 800 IO=1,NO
            EBNO=10.D0**(DEBNO(IO)/10.D0)
            DO 700 IJ=1,NJ
            CALL PUT1(M)
            EBNJ=10.D0**(DEBNJ(IJ)/10.D0)
            RHOO=EBNO*FKK/FLL
            RHOJ=EBNJ*FKK/FLL
            CALL PUT2(LL,DEBNO(IO),DEBNJ(IJ))
            DO 1144 IG=1,16
            IGMA=29-IG
            GM=GAMDAT(IGMA)
            RHOT=GM*RHOO*RHOJ/(RHOO+GM*RHOJ)
            PERR=PEG(GM)
            WRITE(6,1)PERR,GM
      1     FORMAT(1X,'PE= ',1PD12.3, '  GM= ',D12.3)
      1144  CONTINUE
      700   CONTINUE
      800   CONTINUE
            CALL DETACH(6,TERM,IIIIII)
            STOP 0
            END
```



```
            SUBROUTINE GET
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                 C
      C INTERACTIVE INPUT OF PARAMETERS  C
      C                                 C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            COMMON /MARY/M,FM
            COMMON /AK/ FKK
            COMMON /SIZE/ NO, NJ
            DIMENSION RLIST(5)
            DATA RLIST/13.3525,10.6065,9.0939,8.0783,7.3295/
            COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(1)
      20    WRITE(5,21)
      21    FORMAT(' WHAT VALUE OF K? [1]', $)
            READ(5,3,ERR=20)KK
            IF(KK.LT.0.OR.KK.GT.5)GOTO 20
            IF(KK.EQ.0) KK=1
            FKK=KK
            M=2**KK
            RATIO=RLIST(KK)+10.*DLOG10(FKK)
            FM=M
      1     WRITE(5,2)RATIO
      2     FORMAT(' HOW MANY VALUES OF EB/NO? [',F7.4,' DB ONLY] ',$)
            READ(5,3,ERR=1)NO
      3     FORMAT(I2)
            IF(NO.LT.0.OR.NO.GT.10)GOTO 1
            IF(NO.NE.0) THEN
              DO 7 IN=1,NO
      4       WRITE(5,5)IN
      5       FORMAT('  EB/NO(',I2,') (DB) = ',$)
              READ(5,6,ERR=4)DEBNO(IN)
      6       FORMAT(F7.2)
      7       CONTINUE
            ELSE
      C DEFAULT IS RATIO DB ONLY
              NO=1
              DEBNO(1)=RATIO
            END IF
      8     WRITE(5,9)
      9     FORMAT(' HOW MANY VALUES OF EB/NJ? [0(25/7)50 DB] ',$)
            READ(5,3,ERR=8)NJ
            IF(NJ.LT.0.OR.NJ.GT.50)GOTO 8
            IF(NJ.NE.0)THEN
              DO 12 IN=1,NJ
      10      WRITE(5,11)IN
      11      FORMAT('  EB/NJ(',I2,') (DB) = ',$)
              READ(5,6,ERR=10)DEBNJ(IN)
      12      CONTINUE
```

```
PDP-11 FORTRAN-77 V4.0-1        13:40:44    17-Aug-83           Page 3

0044            ELSE
        C SET UP DEFAULT LIST
0045              NJ=15
0046              DO 13 IN=1,NJ
0047              DEBNJ(IN)=25.*(IN-1)/7.
0048    13        CONTINUE
0049            END IF
0050            LLIST(1)=2
0051            RETURN
0052            END


PDP-11 FORTRAN-77 V4.0-1        13:40:50    17-Aug-83           Page 4

0001            SUBROUTINE PUT1(M)
        CCCCCCCCCCCCCCCCCCCCCC
        C                    C
        C WRITE PAGE HEADERS C
        C                    C
        CCCCCCCCCCCCCCCCCCCCCC
0002            WRITE(6,1)M
0003    1       FORMAT('1'//I3,' MFSK/FH WITH AGC USING EXACT EQUATION')
0004            RETURN
0005            END


PDP-11 FORTRAN-77 V4.0-1        13:40:52    17-Aug-83           Page 5

0001            SUBROUTINE PUT2(LL,DEBNO,DEBNJ)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                             C
        C WRITE OUT INPUT PARAMETERS C
        C                             C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            WRITE(6,1)LL,DEBNO,DEBNJ
0004    1       FORMAT(' L = ',I2,' HOPS/BIT',5X,'EB/NO = ',F6.2,' DB',5X,
             $ 'EB/NJ = ',F6.2,' DB')
0005            RETURN
0006            END


PDP-11 FORTRAN-77 V4.0-1        13:40:53    17-Aug-83           Page 6

0001            DOUBLE PRECISION FUNCTION PEG(GM)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                C
        C SUBROUTINE TO CALCULATE P(E) AS FUNCTION OF C
        C                SINGLE ARGUMENT GM              C
        C        WITH INTEGRAND FUNCTION IN GRAND        C
        C                                                C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            EXTERNAL GRAND
0004            DIMENSION WORK(100)
0005            COMMON /HOPS/ LL, FLL
0006            COMMON /MARY/M,FM
0007            COMMON /VARS/ RHOO, RHOJ
0008            COMMON /RHO/ RHO1,RHO2
0009            COMMON /TOTAL/ RHOT
0010            SUMC=0.DO
0011            DO 1000 IL=0,2
0012            IF( IL.EQ.0) THEN
0013              RHO1=RHOO
0014              RHO2=RHOO
0015            ELSE IF(IL.EQ.1) THEN
0016              RHO1=RHOO
0017              RHO2=RHOT
0018            ELSE
0019              RHO1=RHOT
0020              RHO2=RHOT
0021            ENDIF
0022            SUM=0.DO
0023            XL=0.DO
0024    100     XU=XL+2.DO
0025            CALL DGAU20(XL,XU,GRAND,RESULT)
0026              SUM=SUM+RESULT
0027              IF(DABS(RESULT).LE.1.D-4*DABS(SUM))GOTO 200
0028            WRITE(5,1020) XU,SUM
0029    1020    FORMAT(' XU = ',1PE15.8,'   SUM = ',D20.10)
0030            XL=XU
0031            GOTO 100
0032    200     PE=SUM
0033            YL=DBINCO(LL,IL)
0034            SUMC=SUMC+YL*GM**IL*DXI(1.DO-GM,LL-IL)*PE
0035    1000    CONTINUE
0036            PEG=1.DO-SUMC
0037            PEG=FM*PEG*.5DO/(FM-1.DO)
0038            WRITE(5,1040)GM,PEG
0039    1040    FORMAT(' GAMMA = ',1PE15.8,'   PEG = ',E15.8)
0040            RETURN
0041            END
```

```
PDP-11 FORTRAN-77 V4.0-1        13:40:58    17-Aug-83            Page 7

0001             DOUBLE PRECISION FUNCTION GRAND(Z1)
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          C                                            C
          C PRODUCT OF THE PDF OF RANDOM VARIABLE {Z1} C
          C     WITH THE Pr{Zi<Z1}, i=2,...,M          C
          C           Z1---SIGNAL CHANNEL              C
          C           Zi---NOISE CHANNELS              C
          C                                            C
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002             IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003             EXTERNAL GR1,GR2
0004             LOGICAL MORE
0005             COMMON /MARY/ M,FM
0006             COMMON /PASS1/ZZ1
0007             ZZ1=Z1
0008             MORE=.TRUE.
0009             XL=0.D0
0010             SUM=0.D0
0011             SUM1=0.D0
0012      100    XU=XL+5.D0
0013             IF (XU.GT.Z1) THEN
0014               XU=Z1
0015               MORE=.FALSE.
0016             ENDIF
0017             CALL DGAU(XL,XU,GR1,ANS1)
0018             CALL DGAU(XL,XU,GR2,ANS2)
0019             SUM=SUM+ANS1
0020             SUM1=SUM1+ANS2
0021             XL=XU
0022             IF(MORE) GOTO 100
0023             GRAND=SUM*(SUM1**(M-1))
0024             RETURN
0025             END
```

```
PDP-11 FORTRAN-77 V4.0-1        13:41:00    17-Aug-83            Page 8

0001             DOUBLE PRECISION FUNCTION GR1(X)
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          C                            C
          C CONVOLUTION OF TWO RICIANS C
          C                            C
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002             IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003             COMMON /RHO/ RHO1,RHO2
0004             COMMON /PASS1/ZZ1
0005             Y=ZZ1-X
0006             ARG1=X*DSQRT(2.D0*RHO1)
0007             ARG2=Y*DSQRT(2.D0*RHO2)
0008             CALL DXBESI(ARG1,0,ANS1,KODE)
0009             IF(KODE.NE.0)WRITE(5,111)KODE
0010      111    FORMAT(' GR1 FIRST CALL TO DXBESI CODE = ',I1)
0011             CALL DXBESI(ARG2,0,ANS2,KODE)
0012             IF(KODE.NE.0)WRITE(5,112)KODE
0013      112    FORMAT(' GR1 SECOND CALL TO DXBESI CODE = ',I1)
0014             ONE=X*(DEXP(ARG1-RHO1-X**2/2.D0)*ANS1)
0015             TWO=Y*(DEXP(ARG2-RHO2-Y**2/2.D0)*ANS2)
0016             GR1=ONE*TWO
0017             RETURN
0018             END
```

```
PDP-11 FORTRAN-77 V4.0-1        13:41:03    17-Aug-83            Page 9

0001             DOUBLE PRECISION FUNCTION GR2(Y)
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          C                                               C
          C COMPUTE PR{Zi<Z1} WHERE Z1 IS DECISION VARIABLE C
          C   FOR THE SIGNAL CHANNEL AND Zi ARE DECISION    C
          C        VARIABLES FOR NOISE CHANNELS           C
          C               (i=2,...,M)                     C
          C                                               C
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002             IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003             DATA PIR/1.772453850905516027298167D0/
0004             B=PIR*DERF(Y/2.D0)*DEXP(-Y*Y/4.D0)
0005             C=Y*Y/4.D0-.5D0
0006             GR2=B*C+Y/2.D0*DEXP(-Y*Y/2.D0)
0007             RETURN
0008             END
```

```
PDP-11 FORTRAN-77 V4.0-1         13:41:06    17-Aug-83          Page 10

0001            DOUBLE PRECISION FUNCTION DBINCO(N,K)
          CCCCCCCCCCCCCCCCCCCCCCCCCCC
          C                         C
          C  BINOMIAL COEFFICIENTS  C
          C                         C
          CCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            IF(K.GT.N) GOTO 6
0004            IF(K.EQ.N .OR. K.EQ.0)  GOTO1
0005            IF(K.EQ.1 .OR. K.EQ.N-1) GOTO2
0006            IF(K.GT.N/2) GOTO3
0007            KK=K
0008      5     C=N+1
0009            A=N
0010            DO 4 J=2,KK
0011            A=A*(C-J)/J
0012      4     CONTINUE
0013            DBINCO=A
0014            GOTO7
0015      3     KK=N-K
0016            GOTO 5
0017      2     DBINCO=N
0018            GOTO 7
0019      1     DBINCO=1.DO
0020            GOTO 7
0021      6     DBINCO=0.DO
0022      7     RETURN
0023            END
```

```
PDP-11 FORTRAN-77 V4.0-1         13:41:09    17-Aug-83          Page 11

0001            SUBROUTINE DGAU(A,B,F,ANSWER)
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          C                                                         C
          C 20-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL C
          C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4 C
          C                                                         C
          C R. H. FRENCH, 21 JUNE 1983                              C
          C                                                         C
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            DIMENSION X(10),W(10)
0004            DATA X/ 0.076526521133497333755DO,
          1             0.227785851141645078080DO,
          1             0.373706088715419560673DO,
          1             0.510867001950827098004DO,
          1             0.636053680726515025453DO,
          1             0.746331906460150792614DO,
          1             0.839116971822218823395DO,
          1             0.912234428251325905868DO,
          1             0.963971927277913791268DO,
          1             0.993128599185094924786DO /
0005            DATA W/ 0.152753387130725850698DO,
          1             0.149172986472603746788DO,
          1             0.142096109318382051329DO,
          1             0.131688638449176626898DO,
          1             0.118194531961518417312DO,
          1             0.101930119817240435037DO,
          1             0.083276741576704748725DO,
          1             0.062672048334109063570DO,
          1             0.040601429800386941331DO,
          1             0.017614007139152118312DO /
0006            ANSWER=0.DO
0007            BMAO2=(B-A)/2.DO
0008            BPAO2=(B+A)/2.DO
0009            DO 10 I=1,10
0010            C=X(I)*BMAO2
0011            Y1=BPAO2+C
0012            Y2=BPAO2-C
0013            ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0014      10    CONTINUE
0015            ANSWER=ANSWER*BMAO2
0016            RETURN
0017            END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C  SUBROUTINE DXBESI                                                  C
C                                                                     C
C  PURPOSE:                                                           C
C     COMPUTE THE EXP(-X) * I BESSEL FUNCTION FOR A GIVEN ARGUMENT    C
C     AND ORDER IN DOUBLE PRECISION ... RHF, 7 DEC 82                 C
C     ALL NON-INTEGER ARGUMENTS MUST BE DOUBLE PRECISION .. RHF       C
C                                                                     C
C  USAGE:     CALL DXBESI(X,N,BI,IER)                                 C
C                                                                     C
C  DESCRIPTION OF PARAMETERS:                                         C
C           X  -THE ARGUMENT OF THE I BESSEL FUNCTION DESIRED         C
C           N  -THE ORDER OF THE I BESSEL FUNCTION DESIRED            C
C           BI -THE RESULTANT I BESSEL FUNCTION * EXP(-X)             C
C           IER-RESULTANT ERROR CODE WHERE                            C
C              IER=0 NO ERROR                                         C
C              IER=1 N IS NEGATIVE                                    C
C              IER=2 X IS NEGATIVE                                    C
C              IER=3 UNDERFLOW, BI .LT. 1.D-38, BI SET TO 0.0D0       C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE DXBESI(X,N, BI,IER)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER*4 K
      DATA TWOPI/6.2831853071795864 6D0/,TOL/1.D-13/
C
C     CHECK FOR ERRORS IN N AND X AND EXIT IF ANY ARE PRESENT
C
      IER=0
      BI=1.0D0
      IF(N)150,15,10
   10 IF(X)160,18,20
   15 IF(X)160,17,20
   17 RETURN
C
C IF N>0 AND X=0 THEN RESULT IS 0.D0
C
   18 BI=0.D0
      RETURN
C
C     IF ARGUMENT GT 12 AND GT N, USE ASYMPTOTIC FORM
C
   20   IF(X-12.D0)40,40,30
   30 IF(X-N)40,40,110
C
C     COMPUTE FIRST TERM OF SERIES AND SET INITIAL VALUE OF THE SUM
C
   40 XX=X/2.D0
   50 TERM=1.0D0
      IF(N) 70,70,55
```



```
   55 DO 60 I=1,N
      FI=I
      IF(DABS(TERM)-1.D-36)56,60,60
   56 IER=3
      BI=0.0D0
      RETURN
   60 TERM=TERM*XX/FI
   70 BI=TERM
      XX=XX*XX
C
C     COMPUTE TERMS, STOPPING WHEN ABS(TERM) LE ABS(SUM OF TERMS)
C     TIMES TOLERANCE
C
      DO 90 K=1,1000
      IF(DABS(TERM)-DABS(BI*TOL))100,100,80
   80 FK=K*(N+K)
      TERM=TERM*(XX/FK)
   90 BI=BI+TERM
C
C     RETURN BI*DEXP(-X) AS ANSWER FROM THE SERIES
C
  100 BI=BI*DEXP(-X)
  101 RETURN
C
C     X GT 12 AND X GT N, SO USE ASYMPTOTIC APPROXIMATION
C
  110 FN=4*N*N
  115 XX=0.125D0/X
      TERM=1.D0
      BI=1.D0
      DO 130 K=1,30
      IF(DABS(TERM)-DABS(TOL*BI))140,140,120
  120 FK=(2*K-1)**2
      TERM=TERM*XX*(FK-FN)/K
  130 BI=BI+TERM
C
C     SIGNIFICANCE LOST AFTER 30 TERMS, TRY SERIES
C
      GO TO 40
C TAKE OUT THE EXPONENTIAL EXP(X) FROM THE ASYMPTOTIC FORMULA TO
C ACCOMPLISH THE MULTIPLICATION BY EXP(-X) ... RHF 7 DEC 82
  140 BI=BI/DSQRT(TWOPI*X)
      GO TO 101
  150 IER=1
      GO TO 101
  160 IER=2
      GO TO 101
      END
```

APPENDIX 4H

COMPUTER PROGRAM TO COMPUTE BIT ERROR PROBABILITY FOR LINEAR-LAW FH/MFSK RECEIVER WITH AGC USING THE EDGEWORTH SERIES APPROXIMATION (L>2)

The following pages contain a listing of the FORTRAN-77 program used to calculate numerical values for the error probability of a linear-law FH/MFSK receiver with AGC in the presence of partial-band noise jamming for L>2 hops/symbol.

The subroutine GAMMA computes the function $\Gamma(x)$; this routine is supplied from the Scientific Subroutine Package available from Digital Equipment Corporation [19, p. 3-42]. The subroutine DGAU20 is a system-library routine identical, except for the name, to subroutine DGAU contained in Appendix 4G, listing page 11. A listing of subroutine DXI is given in Appendix 4I.

The subroutines ATTACH, DETACH, and BEEP contained in this listing contain system calls specific to the RSX-11M operating system for PDP-11 computers as described in Appendix 1A. Calls to ATTACH and DETACH may be omitted on other systems. The subroutine BEEP (or equivalent) is required only on systems where the FORTRAN I/O routines will not allow control codes to pass unchanged (under RSX-11M, the FORTRAN I/O package changes a bell code to a question mark).

The following constants are used at several locations in this program:

$\pi/2 = 1.570796327$

$\sqrt{\pi/2} = 1.253314137$ = mean of normalized Rician variate

$2-\pi/2 = 0.4292036732$ = variance of normalized Rician variate

$$1/\sqrt{2\pi} = 0.398942280401432677939946 1$$

$$P^{-1}(1-10^{-16}) = 8.22208 \quad \text{where } P(\cdot) \text{ is the Gaussian cdf.}$$

For the polynomial coefficients in PGAUSS, see [4, 26.2.17]. For Edgeworth series coefficients, see Appendix 4B.

```
PDP-11 FORTRAN-77 V4.0-1        11:50:14    16-Aug-83           Page 1

            PROGRAM EDGWRT
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                            C
      C THIS PROGRAM COMPUTES THE ERROR PROBABABILITY FOR L HOPS/SYMBOL C
      C      FH/MFSK RECEIVER EMPLOYING AGC LINEAR LAW DETECTOR    C
      C           USING THE EDGEWORTH SERIES APPROXIMATION         C
      C                                                            C
      C PROGRAMMERS/ANALYSTS:  R.H. FRENCH,  A. KADRICHU,  L.E. MILLER C
      C                                                            C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            LOGICAL*1 TERM
            EXTERNAL PEG
            COMMON /VARS/ RHOO, RHOJ
            COMMON /AK/ FKK
            COMMON /TOTAL/ RHOT
            COMMON /MARY/M,FM
            COMMON /HOPS/ LL, FLL
            COMMON /SIZE/ NO, NJ, NL
            COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(5)
            DIMENSION GAMDAT(28)
            DATA GAMDAT/.001D0,.002D0,.003D0,.004D0,.005D0, .006D0,
         $            .007D0,.008D0,.009D0,.01D0,.02D0 ,.03D0,.04D0,
         $            .05D0, .06D0, .07D0, .08D0,.09D0,.1D0,.2D0,
         $            .3D0, .4D0,.5D0,.6D0,.7D0,.8D0,.9D0,1.D0/
            CALL ATTACH(6,TERM,IIIIII)
            CALL GET
            DO 900 IL=1,NL
            LL=LLIST(IL)
            FLL=LL
            DO 800 IO=1,NO
            EBNO=10.D0**(DEBNO(IO)/10.D0)
            DO 700 IJ=1,NJ
            CALL PUT1(M)
            EBNJ=10.D0**(DEBNJ(IJ)/10.D0)
            RHOO=EBNO*FKK/FLL
            RHOJ=EBNJ*FKK/FLL
            CALL PUT2(LL,DEBNO(IO),DEBNJ(IJ))
            DO 1144 IG=7,11
            IGMA=29-IG
            GM=GAMDAT(IGMA)
            RHOT=GM*RHOO*RHOJ/(RHOO+GM*RHOJ)
            PERR=PEG(GM)
            CALL BEEP(5)
            WRITE(6,1)PERR,GM
            WRITE(5,1)PERR,GM
      1     FORMAT(1X,'PE= ',1PD12.3, '  GM= ',D12.3)
      1144  CONTINUE
      700   CONTINUE
      800   CONTINUE
      900   CONTINUE
            CALL DETACH(6,TERM,IIIIII)
            STOP 0
            END
```

```
PDP-11 FORTRAN-77 V4.0-1        11:50:19    16-Aug-83           Page 2

            SUBROUTINE GET
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                   C
      C INTERACTIVE INPUT OF PARAMETERS   C
      C                                   C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            COMMON /MARY/M,FM
            COMMON /AK/ FKK
            COMMON /SIZE/ NO, NJ, NL
            DIMENSION RLIST(5)
            DATA RLIST/13.3525,10.6065,9.0939,8.0783,7.3295/
            COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(5)
      20    WRITE(5,21)
      21    FORMAT(' WHAT VALUE OF K? [1]', $)
            READ(5,3,ERR=20)KK
            IF(KK.LT.0.OR.KK.GT.5)GOTO 20
            IF(KK.EQ.0) KK=1
            FKK=KK
            M=2**KK
            RATIO=RLIST(KK)
            FM=M
      1     WRITE(5,2)RATIO
      2     FORMAT(' HOW MANY VALUES OF EB/NO? [',F7.4,' DB ONLY] ',$)
            READ(5,3,ERR=1)NO
      3     FORMAT(I2)
            IF(NO.LT.0.OR.NO.GT.10)GOTO 1
            IF(NO.NE.0) THEN
            DO 7 IN=1,NO
      4     WRITE(5,5)IN
      5     FORMAT('   EB/NO(',I2,') (DB) = ',$)
            READ(5,6,ERR=4)DEBNO(IN)
      6     FORMAT(F7.2)
      7     CONTINUE
            ELSE
      C DEFAULT IS RATIO DB ONLY
            NO=1
            DEBNO(1)=RATIO
            END IF
      8     WRITE(5,9)
      9     FORMAT(' HOW MANY VALUES OF EB/NJ? [0(25/7)50 DB] ',$)
            READ(5,3,ERR=8)NJ
            IF(NJ.LT.0.OR.NJ.GT.50)GOTO 8
            IF(NJ.NE.0)THEN
            DO 12 IN=1,NJ
      10    WRITE(5,11)IN
      11    FORMAT('   EB/NJ(',I2,') (DB) = ',$)
            READ(5,6,ERR=10)DEBNJ(IN)
      12    CONTINUE
            ELSE
      C SET UP DEFAULT LIST
            NJ=15
```

PDP-11 FORTRAN-77 V4.0-1 11:50:19 16-Aug-83 Page 3

```
              DO 13 IN=1,NJ
              DEBNJ(IN)=25.*(IN-1)/7.
   13         CONTINUE
              END IF
   14         WRITE(5,15)
   15         FORMAT(' HOW MANY VALUES OF L? [1,2,3,4,6] ',$)
              READ(5,3,ERR=14)NL
              IF(NL.LT.0.OR.NL.GT.5)GOTO 14
              IF(NL.NE.0)THEN
              DO 19 IN=1,NL
   16         WRITE(5,17)IN
   17         FORMAT('  L(',I2,') = ',$)
              READ(5,18,ERR=16)LLIST(IN)
   18         FORMAT(I5)
   19         CONTINUE
              ELSE
              NL=5
              LLIST(1)=1
              LLIST(2)=2
              LLIST(3)=3
              LLIST(4)=4
              LLIST(5)=6
              END IF
              RETURN
              END
```

PDP-11 FORTRAN-77 V4.0-1 11:50:28 16-Aug-83 Page 4

```
              SUBROUTINE PUT1(M)
       CCCCCCCCCCCCCCCCCCCCCC
       C                    C
       C WRITE PAGE HEADERS C
       C                    C
       CCCCCCCCCCCCCCCCCCCCCC
              WRITE(6,1)M
    1         FORMAT('1'//I3,' MFSK/FH WITH AGC USING EDGEWORTH SERIES')
              RETURN
              END
```

PDP-11 FORTRAN-77 V4.0-1 11:50:29 16-Aug-83 Page 5

```
              SUBROUTINE PUT2(LL,DEBNO,DEBNJ)
       CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       C                            C
       C WRITE OUT INPUT PARAMETERS C
       C                            C
       CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              WRITE(6,1)LL,DEBNO,DEBNJ
    1         FORMAT(' L = ',I2,' HOPS/BIT',5X,'EB/NO = ',F6.2,' DB',5X,
          $ 'EB/NJ = ',F6.2,' DB')
              RETURN
              END
```

PDP-11 FORTRAN-77 V4.0-1 11:50:31 16-Aug-83 Page 6

```
              DOUBLE PRECISION FUNCTION PEG(GM)
       CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       C                                                        C
       C THIS SUBROUTINE CALCULATES P(E) AS FUNCTION OF SINGLE  C
       C        ARGUMENT GM  WITH INTEGRAND FUNCTION            C
       C                DEFINED IN GRAND(X)                     C
       C                                                        C
       CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              EXTERNAL GRAND
              DIMENSION WORK(100)
              COMMON /CONSTS/ C(16),CK(15)
              COMMON /HOPS/ LL, FLL
              COMMON /MARY/M,FM
              COMMON /PARAM/ LS, XMEAN
              COMMON /VARS/ RHOO, RHOJ
              COMMON /TOTAL/ RHOT
              COMMON /SD/ SDVRC
              SUMC=0.DO
              DO 1000 IL=0,LL
              SUM=0.DO
              LS=IL
              XMEAN=LS*1.253314137DO*ONEF1(-0.5DO,1.0DO,-RHOT)+(LL-LS)*
          1  1.253314137DO*ONEF1(-0.5DO,1.0DO,-RHOO)
              SDVRC=SQRT(LS*(2.DO*(1.DO+RHOT)-1.570796327DO*
          1 ONEF1(-0.5DO,1.0DO,-RHOT)**2)+(LL-LS)*(2.DO*(1.DO+RHOO)
          1 -1.570796327DO*ONEF1(-0.5DO,1.0DO,-RHOO)**2))
              CALL CALCC(C)
              CALL CALCK(CK)
              CRITIC=XMEAN-1.253314137DO*LL
              XL=0.DO
  100         XU=XL+1.DO
              CALL DGAU20(XL,XU,GRAND,RESULT)
                SUM=SUM+RESULT
                IF(XU.GT.CRITIC.AND.DABS(RESULT).LE.1.D-5*DABS(SUM))GOTO 200
              XL=XU
              GOTO 100
  200         PE=SUM
              YL=DBINCO(LL,IL)
              SUMC=SUMC+YL*GM**IL*DXI(1.DO-GM,LL-IL)*PE
 1000         CONTINUE
              PEG=SUMC
              PEG=FM*PEG*.5DO/(FM-1.DO)
              RETURN
              END
```

PDP-11 FORTRAN-77 V4.0-1 11:50:36 16-Aug-83 Page 7

```
      DOUBLE PRECISION FUNCTION GRAND(X)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                     C
C THIS INTEGRAND FUNCTION CONTAINS THE PRODUCTS OF TWO C
C     EDGEWORTH SERIES REPRESENTING SIGNAL CHANNEL     C
C              AND (M-1) NOISE CHANNELS                C
C                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION ZD1(17),ZD2(16)
      COMMON /CONSTS/ C(13),CK(12)
      COMMON /PARAM/ LS, XMEAN
      COMMON /MARY/ M,FM
      COMMON /VARS/ RHOO, RHOJ
      COMMON /TOTAL/ RHOT
      COMMON /SD/ SDVRC
      COMMON /HOPS/ LL, FLL
      CALL CALCC(C)
      CALL CALCK(CK)
      Z1=(X-XMEAN)/SDVRC
      Z2=(X-1.253314137D0*LL)/SQRT(0.4292036732D0*LL)
      CALL ZDER(Z1,12,ZD1,KODE)
      IF(KODE.EQ.1) STOP 991
      CALL ZDER(Z2,11,ZD2,KODE)
      IF(KODE.EQ.1) STOP 992
      SUM2=PGAUSS(Z2)
      SUM1=ZD1(1)
      DO 10 I=2,13
      SUM1=SUM1+C(I)*ZD1(I)
      IF(I.NE.13) SUM2=SUM2+CK(I)*ZD2(I)
10    CONTINUE
      SUM1=SUM1/SDVRC
      GRAND=SUM1*(1.D0-(SUM2**(M-1)))
      RETURN
      END
```

PDP-11 FORTRAN-77 V4.0-1 11:50:40 16-Aug-83 Page 8

```
      SUBROUTINE CALCC(C)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                     C
C COEFFICIENTS OF EDGEWORTH SERIES FOR REPRESENTATIONS C
C     OF THE DENSITY FUNCTION OF SIGNAL CHANNEL        C
C                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION GAMMAN(4),C(13)
      CALL CALGN(GAMMAN)
      C(1)=1.0D0
      C(2)=0.0D0
      C(3)=0.0D0
      C(4)=-GAMMAN(1)/6.D0
      C(5)=GAMMAN(2)/24.D0
      C(6)=-GAMMAN(3)/120.D0
      C(7)=(GAMMAN(1)**2)/72.D0+GAMMAN(4)/720.D0
      C(8)=-GAMMAN(1)*GAMMAN(2)/144.D0
      C(9)=GAMMAN(2)**2/1152.D0+GAMMAN(1)*GAMMAN(3)/720.D0
      C(10)=-GAMMAN(1)**3/1296.D0
      C(11)=GAMMAN(1)**2*GAMMAN(2)/1728.D0
      C(12)=0.D0
      C(13)=GAMMAN(1)**4/31104.D0
      RETURN
      END
```

PDP-11 FORTRAN-77 V4.0-1 11:50:42 16-Aug-83 Page 9

```
      SUBROUTINE CALCK(CK)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                         C
C COEFFICIENTS OF EDGEWORTH SERIES FOR REPRESENTATIONS OF THE C
C     CUMULATIVE DISTRIBUTION FUNCTION OF NOISE CHANNEL    C
C                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION GAMMAK(4),CK(12)
      CALL CALGK(GAMMAK)
      CK(1)=0.D0
      CK(2)=0.D0
      CK(3)=-GAMMAK(1)/6.D0
      CK(4)=GAMMAK(2)/24.D0
      CK(5)=-GAMMAK(3)/120.D0
      CK(6)=(GAMMAK(1)**2)/72.D0+GAMMAK(4)/720.D0
      CK(7)=-GAMMAK(1)*GAMMAK(2)/144.D0
      CK(8)=GAMMAK(2)**2/1152.D0+GAMMAK(1)*GAMMAK(3)/720.D0
      CK(9)=-GAMMAK(1)**3/1296.D0
      CK(10)=GAMMAK(1)**2*GAMMAK(2)/1728.D0
      CK(11)=0.D0
      CK(12)=GAMMAK(1)**4/31104.D0
      RETURN
      END
```

PDP-11 FORTRAN-77 V4.0-1 11:50:45 16-Aug-83 Page 10

```
          SUBROUTINE CALGN(GAMMAN)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                       C
C  CUMULANTS OF DECISION VARIABLE Z1 (SIGNAL CHANNEL)   C
C           REPRESENTED BY EDGEWORTH SERIES             C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION GAMMAN(4),CAP1U(6),CAP1J(6)
      COMMON /HOPS/ LL, FLL
      COMMON /PARAM/ LS, XMEAN
      COMMON /VARS/RHOO,RHOJ
      COMMON /TOTAL/ RHOT
      CALL CAPA1U(RHOO,CAP1U)
      CALL CAPA1U(RHOT,CAP1J)
      DO 10 N=1,4
      GAMMAN(N)=(LS*CAP1J(N+2)+(LL-LS)*CAP1U(N+2))/((LS*CAP1J(2)
     1 +(LL-LS)*CAP1U(2))**((N+2.D0)/2.D0))
10    CONTINUE
      RETURN
      END
```

PDP-11 FORTRAN-77 V4.0-1 11:50:48 16-Aug-83 Page 11

```
          SUBROUTINE CALGK(GAMMAK)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                       C
C  CUMULANTS OF DECISION VARIABLE Z1 (NOISE CHANNELS)   C
C           REPRESENTED BY EDGEWORTH SERIES             C
C                     i=2,...,M                         C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION GAMMAK(4),CAP2(6)
      COMMON /HOPS/ LL, FLL
      CALL CAPPA2(CAP2)
      DO 10 N=1,4
      GAMMAK(N)=(CAP2(N+2)/CAP2(2)**((N+2.D0)/2.D0))*(LL**(-N/2.D0))
10    CONTINUE
      RETURN
      END
```

PDP-11 FORTRAN-77 V4.0-1 11:50:50 16-Aug-83 Page 12

```
          SUBROUTINE CAPA1U(RHOO,CAP1U)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                        C
C  CUMULANTS OF NORMALIZED RICIAN RANDOM VARIABLE (Z1k)  C
C    FOR NO-JAMMING SITUATION CAPA1U(RHOO,CAP1U) AND     C
C      FOR JAMMING SITUATION CAPA1U(RHOT,CAP1J)          C
C            REPRESENTED BY EDGEWORTH SERIES             C
C                    (k=1,2,...,L)                       C
C                                                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION CAP1U(6)
      X1=ONEF1(-0.5D0,1.0D0,-RHOO)
      X2=ONEF1(-1.5D0,1.0D0,-RHOO)
      X3=ONEF1(-2.5D0,1.0D0,-RHOO)
      X4=ONEF1(-3.5D0,1.0D0,-RHOO)
      CAP1U(1)=1.253314137D0*X1
      CAP1U(2)=2.D0*(1.D0+RHOO)-CAP1U(1)**2
      CAP1U(3)=3.759942412D0*X2
     1 -3*CAP1U(1)*CAP1U(2)-DXI(CAP1U(1),3)
      CAP1U(4)=8.D0*(1+2.D0*RHOO+RHOO*RHOO/2.D0)-4.D0*CAP1U(1)*CAP1U(3)
     1 -3.D0*CAP1U(2)**2-6.D0*CAP1U(1)**2*CAP1U(2)-CAP1U(1)**4
      CAP1U(5)=18.79971206D0*X3-5.D0*CAP1U(1)*
     1 CAP1U(4)-10.D0*CAP1U(2)*CAP1U(3)-10.D0*CAP1U(1)**2*CAP1U(3)
     1  -15.D0*CAP1U(1)*CAP1U(2)**2-10.D0*CAP1U(1)**3*CAP1U(2)
     1  -CAP1U(1)**5
      CAP1U(6)=48.D0*(1.D0+3.D0*RHOO+1.5D0*RHOO*RHOO+RHOO/6.D0*RHOO
     1  *RHOO)-6.D0*CAP1U(1)*CAP1U(5)
     1 -15.D0*CAP1U(2)*CAP1U(4)-10.D0*CAP1U(3)**2-15.D0*CAP1U(1)
     1  **2*CAP1U(4)-60.D0*CAP1U(1)*CAP1U(2)*CAP1U(3)-15.D0*CAP1U(2)
     1  **3-20.D0*CAP1U(1)**3*CAP1U(3)-45.D0*CAP1U(1)**2*CAP1U(2)**2
     1  -15.D0*CAP1U(1)**4*CAP1U(2)-CAP1U(1)**6
      RETURN
      END
```

```
              SUBROUTINE CAPPA2(CAP2)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                          C
      C  CUMULANTS OF NORMALIZED RAYLEIGH RANDOM VARIABLE {Zik}   C
      C       FOR NO-JAMMING AND/OR JAMMING SITUATION             C
      C            REPRESENTED BY EDGEWORTH SERIES                C
      C               (k=1,2,...,L; i=2,...,M)                    C
      C                                                          C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              DOUBLE PRECISION CAP2(6)
              CAP2(1)=1.253314137D0
              CAP2(2)=2.D0-CAP2(1)**2
              CAP2(3)=3.759942411D0-3.D0*CAP2(1)*CAP2(2)-CAP2(1)**3
              CAP2(4)=8.D0-4.D0*CAP2(1)*CAP2(3)-3.D0*CAP2(2)**2-6.D0*
           1  CAP2(1)**2*CAP2(2)-CAP2(1)**4
              CAP2(5)=18.79971206D0-5.D0*CAP2(1)*CAP2(4)-10.D0*CAP2(2)*CAP2(3)
           1  -10.D0*CAP2(1)**2*CAP2(3)-15.D0*CAP2(1)*CAP2(2)**2
           1  -10.D0*CAP2(1)**3*CAP2(2)-CAP2(1)**5
              CAP2(6)=48.D0-6.D0*CAP2(1)*CAP2(5)-15.D0*CAP2(2)*CAP2(4)-
           1  10.D0*CAP2(3)**2-15.D0*CAP2(1)**2*CAP2(4)-60.D0*CAP2(1)*
           1  CAP2(2)*CAP2(3)-15.D0*CAP2(2)**3-20.D0*CAP2(1)**3*CAP2(3)
           1  -45.D0*CAP2(1)**2*CAP2(2)**2-15.D0*CAP2(1)**4*CAP2(2)
           1  -CAP2(1)**6
              RETURN
              END
```



```
              SUBROUTINE ZDER(X,N,ZARRAY,KODE)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                         C
      C                                       (N)               C
      C COMPUTE Z(X), Z'(X), Z"(X), ...., Z      (X) WHERE      C
      C         Z(X) IS THE GAUSSIAN DENSITY                    C
      C                                                         C
      C KODE IS ERROR RETURN CODE: 0 = NO ERROR                 C
      C                            1 = ARGUMENT N < 0           C
      C                                                         C
      C ZARRAY IS OUTPUT.       ZARRAY(1) = Z(X)                C
      C                         ZARRAY(2) = Z'(X)               C
      C                                 .                       C
      C                                 .                       C
      C                                 .                       C
      C                                      (N)                C
      C                         ZARRAY(N+1) = Z    (X)          C
      C                                                         C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              DOUBLE PRECISION ZARRAY(1)
              KODE=1
              IF(N.LT.0)THEN
              RETURN
              END IF
              KODE=0
              ZARRAY(1)=ZGAUSS(X)
              IF(N.EQ.0)THEN
              RETURN
              END IF
              ZARRAY(2)=-X*ZARRAY(1)
              IF(N.EQ.1)THEN
              RETURN
              END IF
              DO 10 I=2,N
              FM=I-1
              ZARRAY(I+1)=-(X*ZARRAY(I)+FM*ZARRAY(I-1))
        10    CONTINUE
              RETURN
              END
```

```
0001            DOUBLE PRECISION FUNCTION ZGAUSS(X)
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          C                                              C
          C DOUBLE PRECISION GAUSSIAN DENSITY FUNCTION    C
          C                                              C
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            DATA OOSR2P/0.398942280401432677939946 1D0/
0004            DATA XCON/-0.5D0/
0005            ZGAUSS=OOSR2P*DEXP(XCON*X*X)
0006            RETURN
0007            END
```



```
0001            DOUBLE PRECISION FUNCTION PGAUSS(X)
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          C                                                   C
          C DOUBLE PRECISION GAUSSIAN DISTRIBUTION FUNCTION    C
          C                                                   C
          C METHOD: POWER SERIES OR ASYMPTOTIC FORMULA,        C
          C         DEPENDING ON VALUE OF ARGUMENT             C
          C                                                   C
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            DATA OOSR2P/0.3989422804014326779399461D0/
0004            DATA B1,B2,B3,B4,B5,P/0.319381530D0, -0.356563782D0,
              $ 1.781477937D0, -1.821255978D0, 1.330274429D0, 0.2316419D0/
0005            Y=DABS(X)
          C ASYMPTOTIC RANGE?
0006            IF(Y.GE.8.22208D0)GOTO 300
          C NO...
0007            IF(Y.LT.6.D0) THEN
          C USE SERIES WHERE POSSIBLE, FOR ACCURACY...
0008              PART=Y
0009              SUM=PART
0010              A=Y*Y/2.D0
0011              DO 100 N=1,32766
0012              FN=N
0013              PART=-PART*(A/FN)
0014              TERM=PART/(FN+FN+1.D0)
0015              SUM=SUM+TERM
0016              IF(DABS(TERM).LT.DABS(SUM)*1.D-16)GOTO 200
0017      100     CONTINUE
0018      200     IF(X.GE.0.D0)THEN
0019                PGAUSS=0.5D0+OOSR2P*SUM
0020              ELSE
0021                PGAUSS=0.5D0-OOSR2P*SUM
0022              ENDIF
0023            ELSE
          C USE POLYNOMIAL APPROXIMATION WHEN SERIES HAS ROUNDOFF PROBLEMS
0024              T=1.D0/(1.D0+P*DABS(X))
0025              F=ZGAUSS(DABS(X))*((B1+(B2+(B3+(B4+B5*T)*T)*T)*T)*T)
0026              IF(X.GE.0.D0) THEN
0027                PGAUSS=1.D0-F
0028              ELSE
0029                PGAUSS=F
0030              END IF
0031            END IF
0032            RETURN
0033      300   IF(X.GE.0.D0)THEN
          C PGAUSS IS WITHIN 1.D-16 OF 1.0D0, SET TO 1.0D0
0034              PGAUSS=1.D0
0035            ELSE IF(X.LE.-13.D0) THEN
          C UNDERFLOW REGION, SET TO 0.D0
0036              PGAUSS=0.D0
```

```
            ELSE
C USE ASYMPTOTIC FORM FOR MODERATE NEGATIVE ARGUMENTS
              SUM=1.DO
              PART=1.DO
              A=1.DO/(Y*Y)
C USE 7 TERMS OF ASYMPTOTIC SERIES...
              DO 400 N=1,7
              PART=-PART*((N+N-1)*A)
              SUM=SUM+PART
400           CONTINUE
450           PGAUSS=ZGAUSS(Y)*SUM/Y
            ENDIF
            RETURN
            END
```



```
            DOUBLE PRECISION FUNCTION ONEF1(A,B,Z)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                 th C
C COMPUTE 1F1(A,B,Z) NEEDED FOR COMPUTATIONS OF N   C
C        MOMENT FOR RICIAN DISTRIBUTED              C
C               RANDOM VARIABLE                     C
C                                                   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IF(DABS(Z).GT.100.)GOTO 900
      T=1.DO
      S=1.DO
      AM1=A-1.DO
      BM1=B-1.DO
      DO 100 I=1,32766
      C=I
      T=T*(Z/C)
      T=T*((AM1+C)/(BM1+C))
      S=S+T
      IF(DABS(T/S).LT.1.D-8) GOTO200
100   CONTINUE
      WRITE(6,1)A,B,Z
1     FORMAT(' 1F1(',F15.8,',',F15.8,',',F15.8,') NOT EVALUATED TO 10**-
     1 5 ACCURACY')
200   ONEF1=S
      RETURN
900   IF(Z.LT.0.DO)GOTO 901
C USE ASYMPTOTIC FORM FOR POSITIVE ARGUMENTS
      CALL GAMMA(B,GB,KODE)
      IF(KODE.NE.0)WRITE(5,902)KODE
902   FORMAT(' FROM ONEF1: GAMMA FUNCTION ERROR KODE = ',I1)
      CALL GAMMA(A,GA,KODE)
      IF(KODE.NE.0)WRITE(5,902)KODE
      ONEF1=(GB/GA)*DEXP(Z)*Z**(A-B)
      RETURN
901   CALL GAMMA(B,GB,KODE)
      IF(KODE.NE.0)WRITE(5,902)KODE
      CALL GAMMA(B-A,GBA,KODE)
      IF(KODE.NE.0)WRITE(5,902)KODE
      ONEF1=(GB/GBA)/(-Z)**A
      RETURN
      END
```

```
PDP-11 FORTRAN-77 V4.0-1        11:51:14    16-Aug-83           Page 19

              DOUBLE PRECISION FUNCTION DBINCO(N,K)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                 C
      C  COMPUTE BINOMIAL COEFFICIENTS  C
      C                                 C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              IF(K.GT.N) GOTO 6
              IF(K.EQ.N .OR. K.EQ.0)  GOTO1
              IF(K.EQ.1 .OR. K.EQ.N-1) GOTO2
              IF(K.GT.N/2) GOTO3
              KK=K
        5     C=N+1
              A=N
              DO 4 J=2,KK
              A=A*(C-J)/J
        4     CONTINUE
              DBINCO=A
              GOTO7
        3     KK=N-K
              GOTO 5
        2     DBINCO=N
              GOTO 7
        1     DBINCO=1.DO
              GOTO 7
        6     DBINCO=0.DO
        7     RETURN
              END
```

```
PDP-11 FORTRAN-77 V4.0-1        11:51:17    16-Aug-83           Page 20

              SUBROUTINE ATTACH(LUN,TERM,ISUCC)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                      C
      C SUBROUTINE TO ATTACH DEVICE ASSOCIATED WITH A LUN     C
      C PROVIDED THAT DEVICE IS A TERMINAL DEVICE, ELSE       C
      C DO NOTHING                                           C
      C                                                      C
      C METHOD: USE GETLUN SYSTEM CALL TO DETERMINE IF        C
      C         THE DEVICE IS A TERMINAL, THEN IF IT IS      C
      C         USE THE WTQIO SYSTEM CALL TO ISSUE THE       C
      C         ATTACH QIO COMMAND AND WAIT FOR ITS COMPLETION C
      C                                                      C
      C PROGRAMMER: R. H. FRENCH, 14 JULY 1983               C
      C                                                      C
      C USAGE:                                               C
      C       LOGICAL*1 TERM                                 C
      C       CALL ATTACH(LUN,TERM,ISUCC)                    C
      C WHERE                                                C
      C       LUN = LOGICAL UNIT NUMBER OF DEVICE TO ATTACH  C
      C       TERM = RETURNED LOGICAL VALUE, .TRUE. IF THE   C
      C              DEVICE IS A TERMINAL, .FALSE. OTHERWISE C
      C       ISUCC = DIRECTIVE STATUS CODE                  C
      C              1 - SUCCESSFUL COMPLETION               C
      C              ANYTHING ELSE - ERROR OF SOME SORT      C
      C              NOTE: IF DEVICE IS NOT A TERMINAL, THEN C
      C                    ISUCC IS THE DSC FROM GETLUN; ELSE C
      C                    FROM WTQIO                        C
      C RESTRICTIONS:                                        C
      C    1.  THIS CALL MUST APPEAR IN THE PROGRAM BEFORE    C
      C        ANY OTHER I/O STATEMENTS REFERENCING SAME LUN C
      C    2.  THIS PROGRAM USES EVENT FLAG 1                C
      C                                                      C
      C USAGE NOTE: THE VALUE  TERM  IS RETURNED PRIMARILY   C
      C TO PASS TO THE COMPANION ROUTINE DETACH              C
      C                                                      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              LOGICAL*1 TERM
              DIMENSION ISB(2),IWORK(6)
              DATA IOATT/"001400/,IFDTTY/"000004/
              CALL GETLUN(LUN,IWORK,ISUCC)
              TERM=.FALSE.
              ITEST=IAND(IWORK(3),IFDTTY)
              IF(ITEST.NE.IFDTTY) RETURN
              TERM=.TRUE.
              CALL WTQIO(IOATT,LUN,1,,ISB,,ISUCC)
              RETURN
              END
```

```
0001            SUBROUTINE DETACH(LUN,TERM,ISUCC)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                          C
        C SUBROUTINE TO DETACH DEVICE ASSOCIATED WITH A LUN         C
        C PROVIDED THAT DEVICE IS A TERMINAL DEVICE, ELSE           C
        C DO NOTHING                                               C
        C                                                          C
        C METHOD: USE THE PARAMETER TERM (SHOULD BE SET BY ATTACH)  C
        C         TO DETERMINE IF THE DEVICE IS A TERMINAL, THEN IF C
        C         IT IS USE THE WTQIO SYSTEM CALL TO ISSUE THE      C
        C         DETACH QIO COMMAND AND WAIT FOR ITS COMPLETION    C
        C                                                          C
        C PROGRAMMER: R. H. FRENCH, 14 JULY 1983                    C
        C                                                          C
        C USAGE:                                                   C
        C       LOGICAL*1 TERM                                     C
        C       CALL DETACH(LUN,TERM,ISUCC)                        C
        C WHERE                                                    C
        C       LUN = LOGICAL UNIT NUMBER OF DEVICE TO ATTACH       C
        C       TERM = LOGICAL VALUE SET BY CALL TO ATTACH,        C
        C               .TRUE. IF THE DEVICE IS A TERMINAL,        C
        C               .FALSE. OTHERWISE                          C
        C       ISUCC = DIRECTIVE STATUS CODE                      C
        C               1 - SUCCESSFUL COMPLETION                  C
        C               ANYTHING ELSE - ERROR OF SOME SORT         C
        C RESTRICTION: THIS PROGRAM USES EVENT FLAG 1              C
        C                                                          C
        C USAGE NOTE: THE VALUE  TERM  IS USUALLY SET BY A PREVIOUS C
        C               CALL TO THE COMPANION ROUTINE ATTACH       C
        C                                                          C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            LOGICAL*1 TERM
0003            DIMENSION ISB(2),IWORK(6)
0004            DATA IODET/"002000/
0005            IF(TERM) THEN
0006            CALL WTQIO(IOATT,LUN,1,,ISB,,ISUCC)
0007            ELSE
0008            ISUCC=1
0009            END IF
0010            RETURN
0011            END
```



```
0001            SUBROUTINE BEEP(LUN)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                  C
        C  SUBROUTINE TO BEEP TERMINAL BEEPER FROM FORTRAN C
        C                                                  C
        C USAGE...                                         C
        C                                                  C
        C       CALL BEEP(LUN)                             C
        C                                                  C
        C WHERE...                                         C
        C                                                  C
        C       LUN = LOGICAL UNIT NUMBER OF UNIT TO        C
        C             RECEIVE THE BELL CODE                C
        C                                                  C
        C NOTE: SPECIFIC TO RSX-11/M SYSTEMS                C
        C                                                  C
        C R. H. FRENCH,  10 AUGUST 1983                     C
        C                                                  C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            CHARACTER*1 BELL
0003            DIMENSION ISB(2),IPRL(6)
0004            DATA IOWLB/"000400/,BELL/"7/
0005            CALL GETADR(IPRL(1),BELL)
0006            IPRL(2)=1
0007            IPRL(3)=0
0008            IPRL(4)=0
0009            IPRL(5)=0
0010            IPRL(6)=0
0011            CALL WTQIO(IOWLB,LUN,1,,ISB,IPRL,IDS)
0012            RETURN
0013            END
```

APPENDIX 4I

COMPUTER PROGRAM LISTING OF FUNCTION SUBPROGRAM DXI

The following page contains a listing of the FORTRAN-77 function subprogram DXI which is referenced by the programs in several other appendices. The purpose of this subprogram is to avoid error messages and incorrect results when the forms 0^0 and $(-|x|)^n$ are encountered in raising a double precision floating-point number to an integer power. For our computations, we set $0^0 = 1$ and $(-|x|)^n = (-1)^n|x|^n$.

```
      DOUBLE PRECISION FUNCTION DXI(D,I)
C
C FUNCTION TO ALLOW RAISING NEGATIVE OR ZERO
C DOUBLE PRECISION TO INTEGER POWER
C
      DOUBLE PRECISION D,A,S
      A=D
      S=1.D0
      IF(D)10,20,30
10    A=-D
      IF(MOD(I,2).NE.0)S=-1.D0
      GOTO 30
20    DXI=0.D0
      IF(I.EQ.0)DXI=1.D0
      RETURN
30    DXI=S*A**I
      RETURN
      END
```

APPENDIX 5A

COMPUTER PROGRAM TO COMPUTE BIT ERROR PROBABILITY FOR SOUARE-LAW SELF-NORMALIZING FH/BFSK RECEIVER

The following pages contain the listinq for the FORTRAN-77 program used to calculate numerical values for the error probability of a square-law self-normalizing FH/BFSK receiver in the presence of partial-band noise jamming.

For subroutines ATTACH and DETACH, see Appendix 4H, listing pages 20 and 21.

```
PDP-11 FORTRAN-77 V4.0-1          13:51:29    26-Sep-83            Page 1

0001              PROGRAM DOUBLE
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                               C
        C THIS PROGRAM COMPUTES THE ERROR PROBABABILITY FOR L=1,2,3,AND 4 C
        C   HOPS/BIT FH/BFSK SELF-NORMALIZING RECEIVER EMPLOYING SQUARE  C
        C        LAW DETECTOR USING DIRECT NUMERICAL CONVOLUTIONS        C
        C                                                               C
        C SUBROUTINE DGAU2 IS A SECOND COPY OF SUBROUTINE DGAU1 TO AVOID  C
        C     RECURSIVE CALL WHEN DOING DOUBLE INTEGRAL NUMERICALLY       C
        C                                                               C
        C PROGRAMMER :  A. KADRICHU                                       C
        C                                                               C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003              LOGICAL*1 TERM
0004              COMMON/NEW/ A,RHON,RHOT
0005              COMMON /PROM/ FLL, LL, FMM, MM, FKK
0006              COMMON /SIZE/ NO, NJ, NL
0007              COMMON /GAMMA/GM
0008              COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(5)
0009              DIMENSION GAMDAT(51)
0010              DATA GAMDAT/ .000005D0, .000006D0, .000007D0, .000008D0,
           $                  .000009D0, .00001D0,  .00002D0,  .00003D0,
           $                  .00004D0,  .00005D0,  .00006D0,  .00007D0,
           $                  .00008D0,  .00009D0,  .0001D0,   .0002D0,
           $                  .0003D0,   .0004D0,   .0005D0,   .0006D0,
           $                  .0007D0,   .0008D0,   .0009D0,   .001D0,
           $                  .002D0,    .003D0,    .004D0,    .005D0,
           $                  .006D0,    .007D0,    .008D0,    .009D0,
           $                  .01D0,     .02D0 ,    .03D0,     .04D0,
           $                  .05D0,     .06D0,     .07D0,     .08D0,
           $                  .09D0,     .1D0,      .2D0,      .3D0,
           $                  .4D0,      .5D0,      .6D0,      .7D0,
           $                  .8D0,      .9D0,     1.D0 /

0011              CALL ATTACH(6,TERM,IIII)
0012              CALL GET
0013              DO 900 IO=1,NO
0014              EBNO=10.D0**(DEBNO(IO)/10.D0)
0015              DO 950 IL=1,NL
0016              LL=LLIST(IL)
0017              FLL=LL
0018              RHON=EBNO/FLL
0019              DO 800 IJ=1,NJ
0020              CALL PUT1(IO,IJ)
0021              EBNJ=10.D0**(DEBNJ(IJ)/10.D0)
0022              RHOJ=EBNJ/FLL
0023              DO 1144 IG=1,51
0024              IGMA=52-IG
0025              GM=GAMDAT(IGMA)
0026              A=RHOJ/RHON
0027              RHOT=RHON*((GM*A)/(1.D0+(GM*A)))
```

```
PDP-11 FORTRAN-77 V4.0-1          13:51:29    26-Sep-83            Page 2

0028              IF(LL.EQ.1) THEN
0029                PERR=PE1(RHON,RHOT,GM)
0030              ELSE IF(LL.EQ.2) THEN
0031                PERR=PE2(RHON,RHOT,GM)
0032              ELSE IF(LL.EQ.3) THEN
0033                PERR=PE3(RHON,RHOT,GM)
0034              ELSE IF(LL.EQ.4) THEN
0035                PERR=PE4(RHON,RHOT,GM)
0036              ELSE IF(LL.GT.4) THEN
0037                WRITE(6,10000)
0038    10000       FORMAT(' SORRY !!!, TRY FOR LOWER VALUE OF L ')
0039              ENDIF
0040              WRITE(6,1)PERR,GM
0041    1         FORMAT(1X,'PE= ',1PD12.3, '  GM= ',D12.3)
0042    1144      CONTINUE
0043    800       CONTINUE
0044    950       CONTINUE
0045    900       CONTINUE
0046              CALL DETACH(6,TERM,IIII)
0047              STOP
0048              END
```

```
PDP-11 FORTRAN-77 V4.0-1          13:51:37    26-Sep-83            Page 3

0001              DOUBLE PRECISION FUNCTION PE1(RHON,RHOT,GM)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                 C
        C COMPUTING BIT ERROR PROBABILITY FOR L=1 HOP/BIT C
        C                                                 C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003              PE1=GM*.5D0*DEXP(-RHOT/2.D0)+(1.D0-GM)*.5D0*DEXP(-RHON/2.D0)
0004              RETURN
0005              END
```

```
PDP-11 FORTRAN-77 V4.0-1          13:51:47    26-Sep-83          Page 8

0001            DOUBLE PRECISION FUNCTION GRANI2(Z)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                              C
        C INTEGRAND FUNCTION OF THE CONVOLUTION OF THE PDF OF DECISION C
        C    VARIABLE FOR L=1 WITH THE POSITIVE ARGUMENT OF THE PDF    C
        C         OF DECISION VARIABLE FOR L=2                         C
        C                                                              C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            COMMON/PARMS/RHO1,RHO2
0004            COMMON/ARG/XX
0005            G1=.5DO*(1.DO+RHO1/2.DO+RHO1*Z/2.DO)
              1 *DEXP(-RHO1/2.DO+RHO1*Z/2.DO)
0006            Y=XX-Z
0007            G3=.5DO*DEXP(-RHO2+RHO2*Y/2.DO)*(1.DO+RHO2+RHO2**2/6.DO
              1 -(1.DO-RHO2**2/2.0DO)*(Y/2.0DO)-RHO2*(1.DO+RHO2/2)*
              1 (Y/2.DO)**2-(RHO2**2/6)*(Y/2.DO)**3)
0008            GRANI2=G1*G3
0009            RETURN
0010            END
```

```
PDP-11 FORTRAN-77 V4.0-1          13:51:50    26-Sep-83          Page 9

0001            DOUBLE PRECISION FUNCTION PE4(RHON,RHOT,GM)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                   C
        C COMPUTING BIT ERROR PROBABILITY FOR L=4 HOPS/BIT C
        C                                                   C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            EXTERNAL GRAND1,GRAND2,GRAND3
0004            COMMON /PARMS/RHO1,RHO2
0005            RHO2=RHON
0006            CALL DGAU1(-4.DO,0.DO,GRAND1,ANS1)
0007            ANSWER=(1.DO-GM)**4*ANS1
0008            RHO2=RHOT
0009            CALL DGAU1(-4.DO,0.DO,GRAND1,ANS2)
0010            ANSWER=ANSWER+(GM**4)*ANS2
0011            RHO2=RHON
0012            CALL DGAU1(-4.DO,0.DO,GRAND2,ANS3)
0013            ANSWER=ANSWER+4.DO*GM*(1.DO-GM)**3*ANS3
0014            RHO2=RHOT
0015            CALL DGAU1(-4.DO,0.DO,GRAND2,ANS4)
0016            ANSWER=ANSWER+4.DO*(1.DO-GM)*GM**3*ANS4
0017            CALL DGAU1(-4.DO,0.DO,GRAND3,ANS5)
0018            ANSWER=ANSWER+6.DO*(1.DO-GM)**2*GM**2*ANS5
0019            PE=ANSWER
0020            RETURN
0021            END
```

```
PDP-11 FORTRAN-77 V4.0-1          13:51:52    26-Sep-83          Page 10

0001            DOUBLE PRECISION FUNCTION GRAND1(X)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                          C
        C DENSITY FUNCTION OF DECISION VARIABLE DEFINED PIECEWISE C
        C           FOR ALL HOPS JAMMED OR UNJAMMED                C
        C                 FOR 4 HOPS/BIT                           C
        C                                                          C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            EXTERNAL GR1,GR2,GR8
0004            COMMON /ARG1/XXX
0005            XXX=X
0006            IF (X.GT.-2.DO) THEN
0007              CALL DGAU2(-2.DO,X,GR1,ANS1)
0008              CALL DGAU2(X,0.DO,GR2,ANS2)
0009              CALL DGAU2(0.DO,X+2.DO,GR8,ANS3)
0010              ANSWER=ANS1+ANS2+ANS3
0011            ELSE
0012              CALL DGAU2(-2.DO,X+2.DO,GR2,ANS3)
0013              ANSWER=ANS3
0014            ENDIF
0015            GRAND1=ANSWER
0016            RETURN
0017            END
```

```
PDP-11 FORTRAN-77 V4.0-1          13:51:54    26-Sep-83          Page 11

0001            DOUBLE PRECISION FUNCTION GRAND2(X)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                          C
        C DENSITY FUNCTION OF DECISION VARIABLE DEFINED PIECEWISE C
        C           FOR ONE HOP JAMMED OR UNJAMMED                 C
        C                 FOR 4 HOPS/BIT                           C
        C                                                          C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            EXTERNAL GR3,GR4,GR5
0004            COMMON /ARG1/XXX
0005            XXX=X
0006            IF (X.GT.-2.DO) THEN
0007              CALL DGAU2(X,0.DO,GR3,ANS1)
0008              CALL DGAU2(-2.DO,X,GR4,ANS2)
0009              CALL DGAU2(0.DO,X+2.DO,GR5,ANS3)
0010              ANSWER=ANS1+ANS2+ANS3
0011            ELSE
0012              CALL DGAU2(-2.DO,X+2.DO,GR3,ANS4)
0013              ANSWER=ANS4
0014            ENDIF
0015            GRAND2=ANSWER
0016            RETURN
0017            END
```

```
PDP-11 FORTRAN-77 V4.0-1          13:51:56    26-Sep-83          Page 12

0001          DOUBLE PRECISION FUNCTION GRAND3(X)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                        C
        C DENSITY FUNCTION OF DECISION VARIABLE DEFINED PIECEWISE C
        C           FOR TWO HOPS JAMMED OR UNJAMMED              C
        C                 FOR 4 HOPS/BIT                         C
        C                                                        C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003          EXTERNAL GR6,GR7,GR9
0004          COMMON /ARG1/XXX
0005          XXX=X
0006          IF (X.GT.-2.DO) THEN
0007            CALL DGAU2(X,0.DO,GR6,ANS1)
0008            CALL DGAU2(-2.DO,X,GR7,ANS2)
0009            CALL DGAU2(0.DO,X+2.DO,GR9,ANS3)
0010            ANSWER=ANS1+ANS2+ANS3
0011          ELSE
0012            CALL DGAU2(-2.DO,X+2.DO,GR6,ANS3)
0013            ANSWER=ANS3
0014          ENDIF
0015          GRAND3=ANSWER
0016          RETURN
0017          END
```

```
PDP-11 FORTRAN-77 V4.0-1          13:51:58    26-Sep-83          Page 13

0001          DOUBLE PRECISION FUNCTION GR1(Z)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                              C
        C INTEGRAND FUNCTION OF THE CONVOLUTION OF THE NEGATIVE ARGUMENT C
        C   OF THE PDF OF DECISION VARIABLE FOR L=2 WHEN BOTH RHOS ARE   C
        C       THE SAME WITH THE POSITIVE ARGUMENT OF THE PDF           C
        C            OF DECISION VARIABLE FOR L=2 WHEN BOTH              C
        C                   RHOS ARE ALSO THE SAME                       C
        C                                                              C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003          COMMON/PARMS/RHO1,RHO2
0004          COMMON/ARG1/XXX
0005          G2=.5DO*DEXP(-RHO2+RHO2*Z/2.DO)*(1.DO+RHO2+RHO2**2/6.DO
           1  +(1.DO+2*RHO2+RHO2**2/2.0DO)*(Z/2.0DO)+RHO2*(1.DO+RHO2/2)*
           1  (Z/2.DO)**2+(RHO2**2/6)*(Z/2.DO)**3)
0006          Y=XXX-Z
0007          G3=.5DO*DEXP(-RHO2+RHO2*Y/2.DO)*(1.DO+RHO2+RHO2**2/6.DO
           1  -(1.DO-RHO2**2/2.0DO)*(Y/2.0DO)-RHO2*(1.DO+RHO2/2)*
           1  (Y/2.DO)**2-(RHO2**2/6)*(Y/2.DO)**3)
0008          GR1=G2*G3
0009          RETURN
0010          END
```

```
PDP-11 FORTRAN-77 V4.0-1          13:52:01    26-Sep-83          Page 14

0001          DOUBLE PRECISION FUNCTION GR2(Z)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                              C
        C INTEGRAND FUNCTION OF THE CONVOLUTION OF THE NEGATIVE ARGUMENT C
        C   OF THE PDF OF DECISION VARIABLE FOR L=2 WHEN BOTH RHOS ARE   C
        C                  THE SAME WITH ITSELF                          C
        C                                                              C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003          COMMON/PARMS/RHO1,RHO2
0004          COMMON/ARG1/XXX
0005          G2=.5DO*DEXP(-RHO2+RHO2*Z/2.DO)*(1.DO+RHO2+RHO2**2/6.DO
           1  +(1.DO+2*RHO2+RHO2**2/2.0DO)*(Z/2.0DO)+RHO2*(1.DO+RHO2/2)*
           1  (Z/2.DO)**2+(RHO2**2/6)*(Z/2.DO)**3)
0006          Y=XXX-Z
0007          G3=.5DO*DEXP(-RHO2+RHO2*Y/2.DO)*(1.DO+RHO2+RHO2**2/6.DO
           1  +(1.DO+2*RHO2+RHO2**2/2.0DO)*(Y/2.0DO)+RHO2*(1.DO+RHO2/2)*
           1  (Y/2.DO)**2+(RHO2**2/6)*(Y/2.DO)**3)
0008          GR2=G2*G3
0009          RETURN
0010          END
```

```
PDP-11 FORTRAN-77 V4.0-1          13:52:03    26-Sep-83          Page 15

0001          DOUBLE PRECISION FUNCTION GR3(Z)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                              C
        C INTEGRAND FUNCTION OF THE CONVOLUTION OF THE NEGATIVE ARGUMENT C
        C   OF THE PDF OF DECISION VARIABLE FOR L=2 WHEN THE RHOS ARE    C
        C      DIFFERENT WITH THE NEGATIVE ARGUMENT OF THE PDF OF        C
        C             DECISION VARIABLE FOR L=2 WHEN BOTH                C
        C                    RHOS ARE THE SAME                           C
        C                                                              C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003          COMMON/PARMS/RHO1,RHO2
0004          COMMON/ARG1/XXX
0005          COMMON/GAMMA/GM
0006          COMMON/NEW/ A,RHON,RHOT
0007          G4=(1.0DO+GM*A)**3/(2.DO*RHON**3)*
           1  (DEXP(-RHOT+RHON*Z/2.DO)*
           1  RHON*(-2.DO*RHOT+RHON**2*(1.DO+Z/2.DO)/(1.DO+GM*A))
           1  +DEXP(-RHON+RHOT*Z/2.DO)*RHOT*(2.DO*RHON+RHOT*RHON/
           1  (1.DO+GM*A)*(1.DO+Z/2.DO)))
0008          Y=XXX-Z
0009          G2=.5DO*DEXP(-RHO2+RHO2*Y/2.DO)*(1.DO+RHO2+RHO2**2/6.DO
           1  +(1.DO+2*RHO2+RHO2**2/2.0DO)*(Y/2.0DO)+RHO2*(1.DO+RHO2/2)*
           1  (Y/2.DO)**2+(RHO2**2/6)*(Y/2.DO)**3)
0010          GR3=G4*G2
0011          RETURN
0012          END
```

```
0001            DOUBLE PRECISION FUNCTION GR4(Z)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                            C
        C INTEGRAND FUNCTION OF THE CONVOLUTION OF THE NEGATIVE ARGUMENT  C
        C   OF THE PDF OF DECISION VARIABLE FOR L=2 WHEN THE RHOS ARE    C
        C       DIFFERENT WITH THE POSITIVE ARGUMENT OF THE PDF          C
        C            OF DECISION VARIABLE FOR L=2 WHEN BOTH              C
        C                     RHOS ARE THE SAME                          C
        C                                                            C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            COMMON/PARMS/RHO1,RHO2
0004            COMMON/ARG1/XXX
0005            COMMON/GAMMA/GM
0006            COMMON/NEW/ A,RHON,RHOT
0007            G4=(1.0D0+GM*A)**3/(2.D0*RHON**3)*
            1   (DEXP(-RHOT+RHON*Z/2.D0)*
            1   RHON*(-2.D0*RHOT+RHON**2*(1.D0+Z/2.D0)/(1.D0+GM*A))
            1   +DEXP(-RHON+RHOT*Z/2.D0)*RHOT*(2.D0*RHON+RHOT*RHON/
            1   (1.D0+GM*A)*(1.D0+Z/2.D0)))
0008            Y=XXX-Z
0009            G3=.5D0*DEXP(-RHO2+RHO2*Y/2.D0)*(1.D0+RHO2+RHO2**2/6.D0
            1   -(1.D0-RHO2**2/2.0D0)*(Y/2.0D0)-RHO2*(1.D0+RHO2/2)*
            1   (Y/2.D0)**2-(RHO2**2/6)*(Y/2.D0)**3)
0010            GR4=G4*G3
0011            RETURN
0012            END
```



```
0001            DOUBLE PRECISION FUNCTION GR5(Z)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                            C
        C INTEGRAND FUNCTION OF THE CONVOLUTION OF THE POSITIVE ARGUMENT  C
        C   OF THE PDF OF DECISION VARIABLE FOR L=2 WHEN THE RHOS ARE    C
        C       DIFFERENT WITH THE NEGATIVE ARGUMENT OF THE PDF          C
        C            OF DECISION VARIABLE FOR L=2 WHEN BOTH              C
        C                     RHOS ARE THE SAME                          C
        C                                                            C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            COMMON/PARMS/RHO1,RHO2
0004            COMMON/GAMMA/GM
0005            COMMON/NEW/ A,RHON,RHOT
0006            COMMON/ARG1/XXX
0007            RHO=RHON/(1.D0+GM*A)
0008            G5=(1.0D0+GM*A)**2/(2.D0*RHON)*(DEXP(GM*A/(1.D0+GM*A)*
            1   (Z/2.D0-1.D0)*RHON)*
            1   (RHON-2.0D0*GM*A+RHON*GM*A*(RHON-GM*A)
            1   *Z/2.D0/(1.D0+GM*A)**2)
            1   +DEXP(-RHON+RHON*Z/2.D0)*
            1   (RHON*((GM*A)/(1.D0+GM*A))**2+2.D0*GM*A-
            1   RHON*Z/2.D0*((RHON*GM*A/((1.D0+GM*A)**2))+1.D0)))
0009            Y=XXX-Z
0010            G2=.5D0*DEXP(-RHO2+RHO2*Y/2.D0)*(1.D0+RHO2+RHO2**2/6.D0
            1   +(1.D0+2*RHO2+RHO2**2/2.0D0)*(Y/2.0D0)+RHO2*(1.D0+RHO2/2)*
            1   (Y/2.D0)**2+(RHO2**2/6)*(Y/2.D0)**3)
0011            GR5=G5*G2
0012            RETURN
0013            END
```

```
0001            DOUBLE PRECISION FUNCTION GR6(Z)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                                 C
        C INTEGRAND FUNCTION OF THE CONVOLUTION OF THE NEGATIVE ARGUMENT  C
        C   OF THE PDF OF DECISION VARIABLE FOR L=2 WHEN THE RHOS ARE     C
        C                  DIFFERENT WITH ITSELF                          C
        C                                                                 C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            COMMON/ARG1/XXX
0004            COMMON/GAMMA/GM
0005            COMMON/NEW/ A,RHON,RHOT
0006            G4=(1.0D0+GM*A)**3/(2.D0*RHON**3)*
            1   (DEXP(-RHOT+RHON*Z/2.D0)*
            1   RHON*(-2.D0*RHOT+RHON**2*(1.D0+Z/2.D0)/(1.D0+GM*A))
            1   +DEXP(-RHON+RHOT*Z/2.D0)*RHOT*(2.D0*RHON+RHOT*RHON/
            1   (1.D0+GM*A)*(1.D0+Z/2.D0)))
0007            Y=XXX-Z
0008            G5=(1.0D0+GM*A)**3/(2.D0*RHON**3)*
            1   (DEXP(-RHOT+RHON*Y/2.D0)*
            1   RHON*(-2.D0*RHOT+RHON**2*(1.D0+Y/2.D0)/(1.D0+GM*A))
            1   +DEXP(-RHON+RHOT*Y/2.D0)*RHOT*(2.D0*RHON+RHOT*RHON/
            1   (1.D0+GM*A)*(1.D0+Y/2.D0)))
0009            GR6=G4*G5
0010            RETURN
0011            END
```



```
0001            DOUBLE PRECISION FUNCTION GR7(Z)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                                 C
        C INTEGRAND FUNCTION OF THE CONVOLUTION OF THE NEGATIVE ARGUMENT  C
        C   OF THE PDF OF DECISION VARIABLE FOR L=2 WHEN THE RHOS ARE     C
        C       DIFFERENT WITH THE POSITIVE ARGUMENT OF THE PDF           C
        C           OF DECISION VARIABLE FOR L=2 WHEN THE                 C
        C               RHOS ARE ALSO DIFFERENT                           C
        C                                                                 C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            COMMON/ARG1/XXX
0004            COMMON/GAMMA/GM
0005            COMMON/NEW/ A,RHON,RHOT
0006            G4=(1.0D0+GM*A)**3/(2.D0*RHON**3)*
            1   (DEXP(-RHOT+RHON*Z/2.D0)*
            1   RHON*(-2.D0*RHOT+RHON**2*(1.D0+Z/2.D0)/(1.D0+GM*A))
            1   +DEXP(-RHON+RHOT*Z/2.D0)*RHOT*(2.D0*RHON+RHOT*RHON/
            1   (1.D0+GM*A)*(1.D0+Z/2.D0)))
0007            Y=XXX-Z
0008            G5=(1.0D0+GM*A)**2/(2.D0*RHON)*(DEXP(GM*A/(1.D0+GM*A)*
            1   (Y/2.D0-1.D0)*RHON)*
            1   (RHON-2.0D0*GM*A+RHON*GM*A*(RHON-GM*A)
            1   *Y/2.D0/(1.D0+GM*A)**2)
            1   +DEXP(-RHON+RHON*Y/2.D0)*
            1   (RHON*((GM*A)/(1.D0+GM*A))**2+2.D0*GM*A-
            1   RHON*Y/2.D0*((RHON*GM*A/((1.D0+GM*A)**2))+1.D0)))
0009            GR7=G4*G5
0010            RETURN
0011            END
```

```
0001          DOUBLE PRECISION FUNCTION GR8(Z)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                              C
      C INTEGRAND FUNCTION OF THE CONVOLUTION OF THE POSITIVE ARGUMENT C
      C   OF THE PDF OF DECISION VARIABLE FOR L=2 WHEN BOTH RHOS ARE  C
      C        THE SAME WITH THE NEGATIVE ARGUMENT OF THE PDF         C
      C             OF DECISION VARIABLE FOR L=2 WHEN BOTH            C
      C                   RHOS ARE ALSO THE SAME                      C
      C                                                              C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003          COMMON/ARG1/XXX
0004          COMMON/PARMS/RHO1,RHO2
0005          COMMON/GAMMA/GM
0006          COMMON/NEW/ A,RHON,RHOT
0007          G3=.5D0*DEXP(-RHO2+RHO2*Z/2.D0)*(1.D0+RHO2+RHO2**2/6.D0
            1 -(1.D0-RHO2**2/2.0D0)*(Z/2.0D0)-RHO2*(1.D0+RHO2/2)*
            1 (Z/2.D0)**2-(RHO2**2/6)*(Z/2.D0)**3)
0008          Y=XXX-Z
0009          G2=.5D0*DEXP(-RHO2+RHO2*Y/2.D0)*(1.D0+RHO2+RHO2**2/6.D0
            1 +(1.D0+2*RHO2+RHO2**2/2.0D0)*(Y/2.0D0)+RHO2*(1.D0+RHO2/2)*
            1 (Y/2.D0)**2+(RHO2**2/6)*(Y/2.D0)**3)
0010          GR8=G3*G2
0011          RETURN
0012          END
```



```
0001          DOUBLE PRECISION FUNCTION GR9(Z)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                              C
      C INTEGRAND FUNCTION OF THE CONVOLUTION OF THE POSITIVE ARGUMENT C
      C   OF THE PDF OF DECISION VARIABLE FOR L=2 WHEN THE RHOS ARE   C
      C      DIFFERENT WITH THE NEGATIVE ARGUMENT OF THE PDF          C
      C             OF DECISION VARIABLE FOR L=2 WHEN THE             C
      C                  RHOS ARE ALSO DIFFERENT                      C
      C                                                              C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003          COMMON/ARG1/XXX
0004          COMMON/GAMMA/GM
0005          COMMON/NEW/ A,RHON,RHOT
0006          G5=(1.0D0+GM*A)**2/(2.D0*RHON)*(DEXP(GM*A/(1.D0+GM*A)*
            1 (Z/2.D0-1.D0)*RHON)*
            1 (RHON-2.0D0*GM*A+RHON*GM*A*(RHON-GM*A)
            1 *Z/2.D0/(1.D0+GM*A)**2)
            1 +DEXP(-RHON+RHON*Z/2.D0)*
            1 (RHON*((GM*A)/(1.D0+GM*A))**2+2.D0*GM*A-
            1 RHON*Z/2.D0*((RHON*GM*A/((1.D0+GM*A)**2))+1.D0)))
0007          Y=XXX-Z
0008          G4=(1.0D0+GM*A)**3/(2.D0*RHON**3)*
            1 (DEXP(-RHOT+RHON*Y/2.D0)*
            1 RHON*(-2.D0*RHOT+RHON**2*(1.D0+Y/2.D0)/(1.D0+GM*A))
            1 +DEXP(-RHON+RHOT*Y/2.D0)*RHOT*(2.D0*RHON+RHOT*RHON/
            1 (1.D0+GM*A)*(1.D0+Y/2.D0)))
0009          GR9=G5*G4
0010          RETURN
0011          END
```

```
PDP-11 FORTRAN-77 V4.0-1        13:52:23    26-Sep-83           Page 22

              SUBROUTINE GET
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                  C
        C INTERACTIVE INPUT OF PARAMETERS  C
        C                                  C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              DIMENSION RLIST(5)
              COMMON /PROM/ FLL, LL, FMM, MM, FKK
              COMMON /SIZE/ NO, NJ, NL
              COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(5)
              DATA RLIST/13.3525,10.6065,9.0939,8.0783,7.3295/
        20    WRITE(5,21)
        21    FORMAT(' WHAT VALUE OF K? [1]', $)
              READ(5,3,ERR=20)KK
              IF(KK.LT.0.OR.KK.GT.5)GOTO 20
              IF(KK.EQ.0) KK=1
              FKK=KK
              MM=2**KK
              RATIO=RLIST(KK)+10.*DLOG10(FKK)
              FMM=MM
        1     WRITE(5,2)RATIO
        2     FORMAT(' HOW MANY VALUES OF EB/NO? [',F7.4,' DB ONLY] ',$)
              READ(5,3,ERR=1)NO
        3     FORMAT(I2)
              IF(NO.LT.0.OR.NO.GT.10)GOTO 1
              IF(NO.NE.0) THEN
                DO 7 IN=1,NO
        4       WRITE(5,5)IN
        5       FORMAT('  EB/NO(',I2,') (DB) = ',$)
                READ(5,6,ERR=4)DEBNO(IN)
        6       FORMAT(F7.2)
        7       CONTINUE
              ELSE
        C DEFAULT IS RATIO DB ONLY
                NO=1
                DEBNO(1)=RATIO
              END IF
        8     WRITE(5,9)
        9     FORMAT(' HOW MANY VALUES OF EB/NJ? [0(25/7)50 DB] ',$)
              READ(5,3,ERR=8)NJ
              IF(NJ.LT.0.OR.NJ.GT.50)GOTO 8
              IF(NJ.NE.0)THEN
                DO 12 IN=1,NJ
        10      WRITE(5,11)IN
        11      FORMAT('  EB/NJ(',I2,') (DB) = ',$)
                READ(5,6,ERR=10)DEBNJ(IN)
        12      CONTINUE
```

```
PDP-11 FORTRAN-77 V4.0-1        13:52:23    26-Sep-8            Page 23

              ELSE
        C SET UP DEFAULT LIST
                NJ=15
                DO 13 IN=1,NJ
                DEBNJ(IN)=25.*(IN-1)/7.
        13      CONTINUE
              END IF
        14    WRITE(5,15)
        15    FORMAT(' HOW MANY VALUES OF L? [1,2,3,4,6] ',$)
              READ(5,3,ERR=14)NL
              IF(NL.LT.0.OR.NL.GT.5)GOTO 14
              IF(NL.NE.0)THEN
                DO 19 IN=1,NL
        16      WRITE(5,17)IN
        17      FORMAT('  L(',I2,') = ',$)
                READ(5,18,ERR=16)LLIST(IN)
        18      FORMAT(I5)
        19      CONTINUE
              ELSE
                NL=5
                LLIST(1)=1
                LLIST(2)=2
                LLIST(3)=3
                LLIST(4)=4
                LLIST(5)=6
              END IF
              RETURN
              END
```

```
PDP-11 FORTRAN-77 V4.0-1        13:52:31    26-Sep-83           Page 24

              SUBROUTINE PUT1(IO,IJ)
        CCCCCCCCCCCCCCCCCCCCCCC
        C                     C
        C WRITE PAGE HEADERS  C
        C                     C
        CCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              COMMON /PROM/ FLL, LL, FMM, MM, FKK
              COMMON /SIZE/ NO, NJ, NL
              COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(5)
              WRITE(6,1)DEBNO(IO), MM,LL,DEBNJ(IJ)
        1     FORMAT('1',31X,
            1 'EXACT PERFORMANCE OF SELF NORMALIZING RECEIVER (BINARY CASE)'
            $ /2X,'EB/NO = ',
            $ F6.2,' dB',10X,'M = ',I2,' L = ',I2,' EB/NJ=',F5.2, ' dB'/)
              RETURN
              END
```

```
PDP-11 FORTRAN-77 V4.0-1          13:52:33    26-Sep-83          Page 25

0001            SUBROUTINE DGAU1(A,B,F,ANSWER)
         CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
         C                                                       C
         C 20-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL C
         C                                                       C
         C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4 C
         C                                                       C
         C R. H. FRENCH, 21 JUNE 1983                            C
         C                                                       C
         CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            DIMENSION X(10),W(10)
0004            DATA X/ 0.076526521133497333755D0,
             1          0.227785851141645078080D0,
             1          0.373706088715419560673D0,
             1          0.510867001950827098004D0,
             1          0.636053680726515025453D0,
             1          0.746331906460150792614D0,
             1          0.839116971822218823395D0,
             1          0.912234428251325905868D0,
             1          0.963971927277913791268D0,
             1          0.993128599185094924786D0 /
0005            DATA W/ 0.152753387130725850698D0,
             1          0.149172986472603746788D0,
             1          0.142096109318382051329D0,
             1          0.131688638449176626898D0,
             1          0.118194531961518417312D0,
             1          0.101930119817240435037D0,
             1          0.083276741576704748725D0,
             1          0.062672048334109063570D0,
             1          0.040601429800386941331D0,
             1          0.017614007139152118312D0 /
0006            ANSWER=0.D0
0007            BMAO2=(B-A)/2.D0
0008            BPAO2=(B+A)/2.D0
0009            DO 10 I=1,10
0010            C=X(I)*BMAO2
0011            Y1=BPAO2+C
0012            Y2=BPAO2-C
0013            ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0014     10     CONTINUE
0015            ANSWER=ANSWER*BMAO2
0016            RETURN
0017            END
```

APPENDIX 5B

EQUATIONS USED IN COMPUTER PROGRAM OF APPENDIX 5A

For L=1

$$P_b(e;\gamma) = \gamma \frac{1}{2} e^{-\rho_T/2} + (1-\gamma) \frac{1}{2} e^{-\rho_N/2} \tag{5B-1}$$

For L=2

$$P_b(e;\gamma) = (1-\gamma)^2 \frac{1}{2} e^{-\rho_N/2}\left(1 + \rho_N/6\right) + \gamma^2 \frac{1}{2} e^{-\rho_T/2}\left(1 + \rho_T/6\right)$$

$$+ 2\gamma(1-\gamma) \frac{1}{\left(\rho_N/2 - \rho_T/2\right)^3} \left\{\frac{1}{2}\left[\frac{1}{2}\rho_N\left(\rho_N - \rho_T\right) - \left(\rho_N + \rho_T\right)\right] e^{-\rho_T/2}\right.$$

$$\left. + \frac{1}{2}\left[\rho_T\left(\rho_N - \rho_T\right) + \left(\rho_N + \rho_T\right)\right] e^{-\rho_N/2}\right\} \tag{5B-2}$$

For L=3

$$P_b(e;\gamma) = (1-\gamma)^3 A + 3\gamma^2(1-\gamma) B + 3\gamma(1-\gamma)^2 C + \gamma^3 D \tag{5B-3a}$$

where

$$\left.\begin{aligned} A = & \int_{-3}^{-1} dx \int_{-1}^{x+2} g_1(z,\rho)\, g_2(x-z,\rho)dz \\ & + \int_{-1}^{0} dx \int_{x}^{1} g_1(z,\rho)\, g_2(x-z,\rho)dz \\ & + \int_{-1}^{0} dx \int_{-1}^{x} g_1(z,\rho)\, g_3(x-z,\rho)dz \end{aligned}\right\} \rho = \rho_N \tag{5B-3b}$$

$$\left.\begin{aligned} B = & \int_{-3}^{-1} dx \int_{-1}^{x+2} g_1(z,\rho_1)\, g_2(x-z,\rho_2)dz \\ & + \int_{-1}^{0} dx \int_{x}^{1} g_1(z,\rho_1)\, g_2(x-z,\rho_2)dz \\ & + \int_{-1}^{0} dx \int_{-1}^{x} g_1(z,\rho_1)\, g_3(x-z,\rho_2)dz \end{aligned}\right\} \begin{aligned} \rho_1 &= \rho_N \\ \rho_2 &= \rho_T \end{aligned} \tag{5B-3c}$$

C = same as B for $\rho_1 = \rho_T$ and $\rho_2 = \rho_N$,

with $g_1(z,\rho)$, $g_2(z,\rho)$ and $g_3(z,\rho)$ defined by

$$g_1(z,\rho) = \frac{1}{2}\left(1+ \frac{\rho}{2} + \frac{\rho_z}{2}\right) \exp\left[- \frac{\rho}{2} + \frac{\rho_z}{2}\right] \qquad |z| < 1 \tag{5B-3d}$$

$$g_2(z,\rho) = \frac{1}{2} e^{-\rho + \rho_z/2} \Big\{ 1 + \rho + \rho^2/6 + (1 + 2\rho + \rho^2/2)(z/2) + \rho(1 + \rho/2)(z/2)^2 + (\rho^2/6)(z/2)^3 \Big\} \qquad -2 < z < 0 \tag{5B-3e}$$

$$g_3(z,\rho) = \frac{1}{2} e^{-\rho + \rho_z/2} \Big\{ 1 + \rho + \rho^2/6 - (1 - \rho^2/2)(z/2) - \rho(1 + \rho/2)(z/2)^2 - (\rho^2/6)(z/2)^3 \Big\} \qquad 0 < z < 2 \tag{5B-3f}$$

D = same as A for $\rho = \rho_T$

<u>For L=4</u>

$$P_b(e;\gamma) = (1-\gamma)^4 A' + \gamma^4 B' + \gamma(1-\gamma)^3 C' + \gamma^3(1-\gamma) D' + \gamma^2(1-\gamma)^2 E' \tag{5B-4a}$$

where

$$\left.\begin{aligned} A' = &\int_{-2}^{0} dx \int_{-2}^{x} g_2(z;\rho_2,\rho_2)\, g_3(x-z,\rho_2,\rho_2)dz \\ &+ \int_{-2}^{0} dx \int_{x}^{0} g_2(z;\rho_2,\rho_2)\, g_2(x-z;\rho_2,\rho_2)dz \\ &+ \int_{-4}^{-2} dx \int_{-2}^{x+2} g_2(z;\rho_2,\rho_2)\, g_2(x-z;\rho_2,\rho_2)dz \\ &+ \int_{-2}^{0} dx \int_{0}^{x+2} g_3(z;\rho_2,\rho_2)\, g_2(x-z,\rho_2,\rho_2)dz \end{aligned}\right\} \rho_2 = \rho_N \tag{5B-4b}$$

$B' =$ same as A' for $\rho_2 = \rho_T$

$$\left.\begin{aligned} C' = &\int_{-4}^{-2} dx \int_{-2}^{x+2} g_2'(z;\rho_1,\rho_2)\, g_2(x-z;\rho_1,\rho_1)dz \\ &+ \int_{-2}^{0} dx \int_{2}^{x} g_2'(z;\rho_1,\rho_2)\, g_2(x-z;\rho_1,\rho_1)dz \\ &+ \int_{-2}^{0} dx \int_{2}^{x} g_2'(z;\rho_1,\rho_2)\, g_3(x-z;\rho_1,\rho_1)dz \\ &+ \int_{-2}^{0} dx \int_{0}^{x+2} g_3'(z;\rho_1,\rho_2)\, g_2(x-z;\rho_1,\rho_2)dz \end{aligned}\right\} \begin{aligned}\rho_1 &= \rho_N \\ \rho_2 &= \rho_T\end{aligned} \tag{5B-4c}$$

$D' =$ same as C' for $\rho_1 = \rho_T$ and $\rho_2 = \rho_N$

$$\left.\begin{aligned} E' = &\int_{-4}^{-2} dx \int_{-2}^{x+2} g_2'(z;\rho_1,\rho_2)\, g_2'(x-z;\rho_1,\rho_2)dz \\ &+ \int_{-2}^{0} dx \int_{-2}^{x} g_2'(z;\rho_1,\rho_2)\, g_2'(x-z;\rho_1,\rho_2)dz \\ &+ \int_{-2}^{0} dx \int_{-2}^{x} g_2'(z;\rho_1,\rho_2)\, g_3'(x-z;\rho_1,\rho_2)dz \\ &+ \int_{-2}^{0} dx \int_{0}^{x+2} g_3'(z;\rho_1,\rho_2)\, g_2'(x-z;\rho_1,\rho_2)dz \end{aligned}\right\} \begin{aligned}\rho_1 &= \rho_N \\ \rho_2 &= \rho_T\end{aligned} \tag{5B-4d}$$

with $g_1(z;\rho)$, $g_2(z;\rho,\rho)$, $g_2'(z;\rho_1,\rho_2)$, $g_3(z;\rho,\rho)$ and $g_3(z;\rho_1,\rho_2)$ defined by:

$$g_1(z;\rho) = \frac{1}{2}\left(1 + \frac{1}{2} + \frac{\rho z}{2}\right)\exp\left[-\frac{\rho}{2} + \frac{\rho z}{2}\right] \qquad |z| < 1 \quad \text{(5B-4e)}$$

$$g_2(z;\rho,\rho) = \frac{1}{2}\exp(-\rho + \rho z/2)\Big\{1 + \rho + \rho^2/6 + (1 + 2\rho + \rho^2/2)(z/2)$$

$$+ \rho(1 + \rho/2)(z/2)^2 + (\rho^2/6)(z/2)^3\Big\} \qquad -2 < z < 0 \quad \text{(5B-4f)}$$

$$g_2'(z;\rho_1,\rho_2) = \frac{1}{2(\rho_1-\rho_2)^3}\Big\{\exp(-\rho_2 + \rho_1 z/2)\ \rho_1\left[-2\rho_2 + \rho_1(\rho_1-\rho_2)(1 + z/2)\right]$$

$$+ \exp(-\rho_1 + \rho_2 z/2)\ \rho_2\left[2\rho_1 + \rho_2(\rho_1-\rho_2)(1 + z/2)\right]\Big\}$$

$$-2 < z < 0 \quad \text{(5B-4g)}$$

$$g_3(z;\rho,\rho) = \frac{1}{2}\exp(-\rho + \rho z/2)\Big\{1 + \rho + \rho^2/6 - (1-\rho^2/2)(z/2)$$

$$- \rho(1 + \rho/2)(z/2)^2 - (\rho^2/6)(z/2)^3\Big\} \qquad 0 < z < 2 \quad \text{(5B-4h)}$$

$$g_3'(z;\rho_1,\rho_2) = \frac{1}{2(\rho_1-\rho_2)^3}\Big\{\exp(-\rho_2 + \rho_2 z/2)\ \big[\rho_1^3 - \rho_1^2\rho_2 - 2\rho_1\rho_2 + (\rho_1-\rho_2)$$

$$\cdot (\rho_1^2\rho_2 - \rho_1\rho_2^2 \cdot \rho_2^2)z/2\big] - \exp(-\rho_1 + \rho_1 z/2)\ \big[\rho_2^3 - \rho_1\rho_2^2 - 2\rho_1\rho_2$$

$$+ (\rho_1-\rho_2)(\rho_1^2\rho_2 - \rho_1\rho_2^2 + \rho_1^2)z/2\big]\Big\} \qquad 0 < z < 2 \quad \text{(5B-4i)}$$

APPENDIX 5C

COMPUTER PROGRAM TO COMPUTE BIT ERROR PROBABILITY FOR SQUARE-LAW SELF-NORMALIZING FH/BFSK RECEIVER WITH AN N-LEVEL QUANTIZER

The following pages contain the listing for the FORTRAN-77 program used to calculate numerical values for the error probability of a square-law self-normalizing FH/BFSK receiver with an N-level quantizer in the presence of partial-band noise jamming.

For subroutines ATTACH and DETACH, see Appendix 4H, listing pages 20 and 21.

The default increments for E_b/N_J in dB are chosen to facilitate plotting on a scale of 7 divisions = 5 dB.

PRECAUTION:

Before running this program, please make sure that the arrays in function PEG are at least as large as

```
DIMENSION V(N), WORK1(N*L-1), WORK2(N*L-1)
```

and the calls to CONVLV specify the size of WORK2 as the sixth parameter, i.e.

```
CALL CONVLV(V, NV, WORK1, NW1, WORK2, N*L-1, KODE)
```

where N is the number of quantization levels and L is the number of hops/bit.

```
PDP-11 FORTRAN-77 V4.0-1        11:58:57     9-Sep-83          Page 1

0001            PROGRAM SNCLIP
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                              C
        C THIS PROGRAM COMPUTES THE ERROR PROBABABILITY OF L HOPS/BIT  C
        C  FH/BFSK SELF-NORMALIZING RECEIVER WITH N-LEVEL QUANTIZER    C
        C                                                              C
        C PROGRAMMER: A. KADRICHU                                      C
        C DATE:       SEPTEMBER 9, 1983                                C
        C                                                              C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            LOGICAL*1 TERM
0004            EXTERNAL PEG
0005            COMMON /VARS/ RHOO, RHOJ
0006            COMMON /AK/ FKK
0007            COMMON /TOTAL/ RHOT
0008            COMMON /MARY/M,FM
0009            COMMON /HOPS/ LL, FLL
0010            COMMON /QUINT/ NQ,DNQ
0011            COMMON /SIZE/ NO, NJ, NL
0012            COMMON /NU/ TRESHO
0013            COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(6)
0014            DIMENSION GAMDAT(51)
0015            DATA GAMDAT/ .000005D0, .000006D0, .000007D0, .000008D0,
             $               .000009D0, .00001D0,  .00002D0,  .00003D0,
             $               .00004D0,  .00005D0,  .00006D0,  .00007D0,
             $               .00008D0,  .00009D0,  .0001D0,   .0002D0,
             $               .0003D0,   .0004D0,   .0005D0,   .0006D0,
             $               .0007D0,   .0008D0,   .0009D0,   .001D0,
             $               .002D0,    .003D0,    .004D0,    .005D0,
             $               .006D0,    .007D0,    .008D0,    .009D0,
             $               .01D0,     .02D0 ,    .03D0,     .04D0,
             $               .05D0,     .06D0,     .07D0,     .08D0,
             $               .09D0,     .1D0,      .2D0,      .3D0,
             $               .4D0,      .5D0,      .6D0,      .7D0,
             $               .8D0,      .9D0,     1.D0 /
0016            CALL ATTACH(6,TERM,IIIIII)
0017            CALL GET
0018            DO 900 IL=1,NL
0019            LL=LLIST(IL)
0020            FLL=LL
0021            DO 800 IO=1,NO
0022            EBNO=10.D0**(DEBNO(IO)/10.D0)
0023            DO 700 IJ=1,NJ
0024            CALL PUT1(M)
0025            EBNJ=10.D0**(DEBNJ(IJ)/10.D0)
0026            RHOO=EBNO*FKK/FLL
0027            RHOJ=EBNJ*FKK/FLL
0028            CALL PUT2(LL,DEBNO(IO),DEBNJ(IJ))

PDP-11 FORTRAN-77 V4.0-1        11:58:57     9-Sep-83          Page 2

0029            DO 1144 IGMA=1,51
0030            GM=GAMDAT(IGMA)
0031            RHOT=GM*RHOO*RHOJ/(RHOO+GM*RHOJ)
0032            IF(NQ.EQ.2) THEN
0033              TRES=0.D0
0034              PERR=PEG(GM,TRES)
0035              WRITE(6,1)PERR,GM
0036              WRITE(5,1)PERR,GM
0037    1         FORMAT(1X,'PE= ',1PD12.3, ' GAMMA= ',D12.3)
0038            ELSE
0039              TRES=TRESHO
0040              PERR=PEG(GM,TRES)
0041              WRITE(6,1)PERR,GM
0042              WRITE(5,1)PERR,GM
0043            ENDIF
0044    1144    CONTINUE
0045    700     CONTINUE
0046    1122    CONTINUE
0047    800     CONTINUE
0048    900     CONTINUE
0049            CALL DETACH(6,TERM,IIIIII)
0050            STOP 0
0051            END
```

```
          SUBROUTINE GET
      CCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                          C
      C INPUT DATA INTERACTIVELY C
      C                          C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCC
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          COMMON /MARY/M,FM
          COMMON /AK/ FKK
          COMMON /NU/ TRESHO
          COMMON /SIZE/ NO, NJ, NL
          COMMON /QUINT/ NQ,DNQ
          DIMENSION RLIST(5)
          DATA RLIST/13.3525,10.6065,9.0939,8.0783,7.3295/
          COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(6)
20        WRITE(5,21)
21        FORMAT(' WHAT VALUE OF K? [1]', $)
          READ(5,3,ERR=20)KK
          IF(KK.LT.0.OR.KK.GT.5)GOTO 20
          IF(KK.EQ.0) KK=1
          FKK=KK
          M=2**KK
          RATIO=RLIST(KK)
          FM=M
1         WRITE(5,2)RATIO
2         FORMAT(' HOW MANY VALUES OF EB/NO? [',F7.4,' DB ONLY] ',$)
          READ(5,3,ERR=1)NO
3         FORMAT(I2)
          IF(NO.LT.0.OR.NO.GT.10)GOTO 1
          IF(NO.NE.0) THEN
            DO 7 IN=1,NO
4           WRITE(5,5)IN
5           FORMAT('  EB/NO(',I2,') (DB) = ',$)
            READ(5,6,ERR=4)DEBNO(IN)
6           FORMAT(F7.2)
7           CONTINUE
          ELSE
      C DEFAULT IS RATIO DB ONLY
            NO=1
            DEBNO(1)=RATIO
          END IF
8         WRITE(5,9)
9         FORMAT(' HOW MANY VALUES OF EB/NJ? [0(25/7)50 DB] ',$)
          READ(5,3,ERR=8)NJ
          IF(NJ.LT.0.OR.NJ.GT.50)GOTO 8
          IF(NJ.NE.0)THEN
            DO 12 IN=1,NJ
10          WRITE(5,11)IN
11          FORMAT('  EB/NJ(',I2,') (DB) = ',$)
            READ(5,6,ERR=10)DEBNJ(IN)
```



```
12          CONTINUE
          ELSE
      C SET UP DEFAULT LIST
            NJ=15
            DO 13 IN=1,NJ
            DEBNJ(IN)=25.*(IN-1)/7.
13          CONTINUE
          END IF
14        WRITE(5,15)
15        FORMAT(' HOW MANY VALUES OF L [6]? ',$)
          READ(5,3,ERR=14)NL
          IF(NL.EQ.0)NL=6
          IF(NL.LE.0.OR.NL.GT.6)GOTO 14
          DO 19 IL=1,NL
16        WRITE(5,17)IL,IL
17        FORMAT('  L(',I2,') [',I2,'] = ',$)
          READ(5,18,ERR=16)LLIST(IL)
18        FORMAT(I5)
          IF(LLIST(IL).EQ.0)LLIST(IL)=IL
          IF(LLIST(IL).LE.0.OR.LLIST(IL).GT.6)GOTO 16
19        CONTINUE
          WRITE(5,31)
31        FORMAT(' INPUT LEVEL OF QUANTIZER ',$)
          READ(5,3) NQ
          DNQ=NQ
          WRITE(5,33)
33        FORMAT(' INPUT THRESHOLD VALUE ',$)
          READ(5,34) TRESHO
34        FORMAT(D12.3)
          RETURN
          END
```

```
PDP-11 FORTRAN-77 V4.0-1        11:59:20      9-Sep-83            Page 5

0001            SUBROUTINE PUT1(M)
        CCCCCCCCCCCCCCCCCCCCCC
        C                    C .
        C WRITE PAGE HEADERS C
        C                    C
        CCCCCCCCCCCCCCCCCCCCCC
0002            COMMON /QUINT/ NQ,DNQ
0003            WRITE(6,1)NQ
0004    1       FORMAT('1'//,' BFSK/FH  SELF-NORMALIZING W/',I3,'LEVEL QUANTIZER')
0005            RETURN
0006            END
```

```
PDP-11 FORTRAN-77 V4.0-1        11:59:22      9-Sep-83            Page 6

0001            SUBROUTINE PUT2(L,DEBNO,DEBNJ)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                             C
        C WRITE OUT CURRENT PARAMETERS C
        C                             C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            COMMON /NU/ TRESHO
0004            WRITE(6,1)L,DEBNO,DEBNJ,TRESHO
0005    1       FORMAT(' L = ',I2,4X,'EB/NO = ',F6.2,' DB',4X,
             $ 'EB/NJ = ',F6.2,' DB AND THRESHOLD= ', F7.3)
0006            RETURN
0007            END
```

```
PDP-11 FORTRAN-77 V4.0-1        11:59:24      9-Sep-83            Page 7

0001            DOUBLE PRECISION FUNCTION PEG(GM,TRES)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                    C
        C SUBROUTINE TO CALCULATE P(E) AS A  C
        C FUNCTION OF GM AND THRESHOLD ETA   C
        C                                    C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            COMMON /QUINT/ NQ,DNQ
0004            COMMON /INPUTS/ DEBNO(10), DEBNJ(50), LLIST(6)
0005            COMMON /HOPS/ LL, FLL
0006            COMMON /VARS/ RHOO, RHOJ
0007            COMMON /TOTAL/ RHOT
0008            DIMENSION V(32),WORK1(187),WORK2(187)
0009            SUMC=0.DO
0010            NV=NQ
0011            DO 3333 IL=0,LL
        C " for single hop case "
        C ----------------------
0012            IF(LL.EQ.1) THEN
0013              IF (IL.EQ.0)THEN
0014                RHO=RHOO
0015              ELSE
0016                RHO=RHOT
0017              ENDIF
0018              CALL VALUE(RHO,TRES,NQ,V)
0019              DO 10 I=1,NV
0020    10        WORK2(I)=V(I)
0021              NW2=NV
0022              IUPPER=NW2/2
0023              SUM=0.DO
0024              DO 9989 ISQ=1,IUPPER
0025              SUM=SUM+WORK2(ISQ)
0026    9989      CONTINUE
0027              IF(IUPPER*2.NE.NW2) SUM=SUM+.5DO*WORK2(IUPPER+1)
0028            ELSE
        C " for multiple hops case "
        C -------------------------
0029              NW2=NV
        C         " when some of the hops are jammed and unjammed "
        C         -------------------------------------------------
0030              IF(IL.GE.2.AND.IL.LE.LL-2) THEN
0031                CALL VALUE(RHOT,TRES,NQ,V)
0032                DO 15 J=1,NW2
0033    15          WORK2(J)=V(J)
0034                DO 9100 I=2,IL
0035                DO 20 J=1,NW2
0036    20          WORK1(J)=WORK2(J)
0037                NW1=NW2
0038                CALL CONVLV(V,NV,WORK1,NW1,WORK2,187,KODE)
```

```
0039               NW2=NW1+NV-1
0040   9100        CONTINUE
0041               CALL VALUE(RHOO,TRES,NQ,V)
0042               DO 9101 I=IL+1,LL
0043               DO 201 J=1,NW2
0044   201         WORK1(J)=WORK2(J)
0045               NW1=NW2
0046               CALL CONVLV(V,NV,WORK1,NW1,WORK2,187,KODE)
0047               NW2=NW1+NV-1
0048   9101        CONTINUE
       C        " special cases "
       C         ----------------
0049              ELSE
       C         " when none of the hops are jammed "
       C          ----------------------------------
0050               IF(IL.EQ.0) THEN
0051                 CALL VALUE(RHOO,TRES,NQ,V)
0052                 DO 9102 MOVE=1,NV
0053                 WORK2(MOVE)=V(MOVE)
0054   9102          CONTINUE
0055                 NW2=NV
       C         " when only one hop is jammed "
       C          -----------------------------
0056               ELSE IF(IL.EQ.1) THEN
0057                 CALL VALUE(RHOO,TRES,NQ,V)
0058                 CALL VALUE(RHOT,TRES,NQ,WORK2)
0059                 NW2=NV
       C         " when only one hop is unjammed "
       C          -------------------------------
0060               ELSE IF(IL.EQ.LL-1) THEN
0061                 CALL VALUE(RHOT,TRES,NQ,V)
0062                 CALL VALUE(RHOO,TRES,NQ,WORK2)
0063                 NW2=NV
       C         " when all hops are jammed "
       C          --------------------------
0064               ELSE IF(IL.EQ.LL) THEN
0065                 CALL VALUE(RHOT,TRES,NQ,V)
0066                 DO 9103 MOVE=1,NV
0067                 WORK2(MOVE)=V(MOVE)
0068   9103          CONTINUE
0069                 NW2=NV
0070               ENDIF
       C " all special cases merge here for convolutions "
       C -----------------------------------------------
0071               DO 91010 I=2,LL
0072               DO 2010 J=1,NW2
0073   2010        WORK1(J)=WORK2(J)
0074               NW1=NW2
0075               CALL CONVLV(V,NV,WORK1,NW1,WORK2,187,KODE)
0076               NW2=NW1+NV-1
```



```
0077   91010         CONTINUE
0078               ENDIF
       C " computation of conditional error probability "
       C ----------------------------------------------
0079               IUPPER=NW2/2
0080               SUM=0.DO
0081               DO 9898 ISQ=1,IUPPER
0082               SUM=SUM+WORK2(ISQ)
0083   9898        CONTINUE
0084               IF(IUPPER*2.NE.NW2) SUM=SUM+.5DO*WORK2(IUPPER+1)
0085             ENDIF
       C " computation of unconditional error probability for any L "
       C -----------------------------------------------------------
0086             PE=SUM
0087             YL=DBINCO(LL,IL)
0088             SUMC=SUMC+YL*GM**IL*DXI(1.DO-GM,LL-IL)*PE
0089   3333      CONTINUE
0090             PEG=SUMC
0091             RETURN
0092             END
```

```
              SUBROUTINE VALUE(RHO,TRES,NQ,V)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                            C
      C COMPUTE DISCRETE PROBABILITIES FOR EACH QUANTIZED LEVEL C
      C                                                            C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              DIMENSION V(NQ)
              DNQ=NQ
              DO 12 IQ=1,NQ
              V(IQ)=0.D0
      12      CONTINUE
      C    " for general N-level ( N not equal to 2) "
      C    ----------------------------------------------
              IF (NQ.NE.2) THEN
                IF(TRES.LT.1.D0)THEN
                  V(1)=.5D0*DEXP(-RHO/2.D0)*((1.D0-TRES)*DEXP(RHO/2*(-TRES)))
                  V(NQ)=1.D0-.5*(1+TRES)*DEXP(RHO/2*(TRES-1))
                  DO 103 IQ=2,NQ-1
                  DIQ=IQ
                  A9=DMIN1(1.D0,2.D0*TRES*(DIQ-DNQ/2.D0)/(DNQ-2.D0))
                  A8=DMAX1(-1.D0,2.D0*TRES*(DIQ-1.D0-DNQ/2.D0)/(DNQ-2.D0))
                  V(IQ)=.5D0*DEXP(-RHO/2.D0)*((1.D0+A9)*DEXP(RHO/2*A9)
           $      -(1+A8)*DEXP(RHO/2*A8))
      103         CONTINUE
                ELSE
                  DO 100 IQ=1,NQ
                  DIQ=IQ
                  A9=DMIN1(1.D0,2.D0*TRES*(DIQ-DNQ/2.D0)/(DNQ-2.D0))
                  A8=DMAX1(-1.D0,2.D0*TRES*(DIQ-1.D0-DNQ/2.D0)/(DNQ-2.D0))
                  A8NEXT=DMAX1(-1.D0,2.D0*TRES*(DIQ-DNQ/2.D0)/(DNQ-2.D0))
                  IF(A8.EQ.-1.D0.AND.A8NEXT.EQ.-1.D0) GOTO 100
                  V(IQ)=.5D0*DEXP(-RHO/2.D0)*((1.D0+A9)*DEXP(RHO/2*A9)
           $      -(1+A8)*DEXP(RHO/2*A8))
                  IF(A9.EQ.1.D0) GOTO 101
      100         CONTINUE
                ENDIF
      101       CONTINUE
      C    " for special case N=2 (threshold must be 0) "
      C    ----------------------------------------------
              ELSE IF(NQ.EQ.2 ) THEN
                A9=0.D0
                A8=-1
                V(1)=.5D0*DEXP(-RHO/2.D0)
                V(2)=.5D0*DEXP(-RHO/2.D0)*(2.D0*DEXP(RHO/2.D0)-1.D0)
              ENDIF
              RETURN
              END
```



```
              SUBROUTINE CONVLV(A,NA,B,NB,C,NC,KODE)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                    C
      C SUBROUTINE TO PERFORM THE DISCRETE CONVOLUTION OF TWO ARRAYS        C
      C                                                                    C
      C DEFINITION:                                                        C
      C                                                                    C
      C                     ---  ---                                       C
      C                     \    \                                         C
      C            C(K) =   /    /    A(I) * B(J)                          C
      C                     ---  ---                                       C
      C                     I+J-1=K                                        C
      C                                                                    C
      C USAGE:                                                             C
      C                                                                    C
      C       DIMENSION A( ), B( ), C( )                                   C
      C       CALL CONVLV(A,NA,B,NB,C,NC,KODE)                             C
      C                                                                    C
      C  WHERE                                                             C
      C                                                                    C
      C        A - INPUT ARRAY #1 (DOUBLE PRECISION)                       C
      C       NA - NUMBER OF ELEMENTS IN A WHICH PARTICIPATE IN CONVOLUTION C
      C        B - INPUT ARRAY #2 (DOUBLE PRECISION)                       C
      C       NB - NUMBER OF ELEMENTS IN B WHICH PARTICIPATE IN CONVOLUTION C
      C        C - OUTPUT ARRAY (DOUBLE PRECISION)                         C
      C       NC - SIZE OF ARRAY C  (NC .GE. NA+NB-1)                      C
      C     KODE - ERROR RETURN CODE, DEFINED AS FOLLOWS:                  C
      C               KODE = 0  -  NO ERROR                                C
      C                      1  -  C ARRAY TOO SMALL TO HOLD RESULT        C
      C                      2  -  ONE OR MORE OF NA, NB, NC ARE .LE. 0    C
      C                                                                    C
      C  PROGRAMMER: R. H. FRENCH                                          C
      C              8 SEPTEMBER 1983                                      C
      C                                                                    C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              DIMENSION A(NA),B(NB),C(NC)
              IF(NC.LT.NA+NB-1) THEN
              KODE=1
              ELSE IF(NA.LE.0.OR.NB.LE.0.OR.NC.LE.0) THEN
              KODE=2
              ELSE
              KODE=0
              ENDIF
              IF(KODE.NE.0) RETURN
              DO 10 IC=1,NC
      10      C(IC)=0.D0
              DO 20 IA=1,NA
              DO 20 IB=1,NB
              ISUB=IA+IB-1
      20      C(ISUB)=C(ISUB)+A(IA)*B(IB)
              RETURN
              END
```

```
          DOUBLE PRECISION FUNCTION DBINCO(N,K)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                              C
C DOUBLE PRECISION BINOMIAL COEFFICIENT                        C
C                                                              C
C DEFINITION:                                                  C
C                                                              C
C   IF K .GT. 0  .AND. K .LE. N, THEN  Y = N!/((K!(N-K)!)      C
C   ELSE  Y = 0.                                               C
C                                                              C
C USAGE:                                                       C
C                                                              C
C        Y = DBINCO(N,K)                                       C
C                                                              C
C WHERE                                                        C
C                                                              C
C        Y AND DBINCO ARE DECLARED DOUBLE PRECISION IN CALLING C
C                     PROGRAM.                                 C
C        N AND K ARE INTEGERS.                                 C
C                                                              C
C PROGRAMMER: R. H. FRENCH                                     C
C             CA. 1977                                         C
C                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          IF(K.GT.N) GOTO 6
          IF(K.EQ.N .OR. K.EQ.0)  GOTO1
          IF(K.EQ.1 .OR. K.EQ.N-1) GOTO2
          IF(K.GT.N/2) GOTO3
          KK=K
    5     C=N+1
          A=N
          DO 4 J=2,KK
          A=A*(C-J)/J
    4     CONTINUE
          DBINCO=A
          GOTO7
    3     KK=N-K
          GOTO 5
    2     DBINCO=N
          GOTO 7
    1     DBINCO=1.DO
          GOTO 7
    6     DBINCO=0.DO
    7     RETURN
          END
```

APPENDIX 5D

DERIVATION OF (5-70)

Using (5-50), we can write

$$p_3\left(u,v;\sigma_k^2\right) = \frac{v}{2}\, p_1\left[\frac{1}{2}\, v(1+u);\sigma_k^2\right]\, p_2\left[\frac{1}{2}\, v(1-u);\sigma_k^2\right] \tag{5D-1}$$

where

$$p_1(\alpha) = \frac{\alpha}{\sigma^2} \exp\left(-\frac{A^2+\alpha^2}{2\sigma^2}\right) I_0\left(\frac{A\alpha}{\sigma^2}\right) \tag{5D-2a}$$

and

$$p_2(\beta) = \frac{\beta}{\sigma^2} \exp\left(-\frac{\beta^2}{2\sigma^2}\right). \tag{5D-2b}$$

Substituting (5D-2) into (5D-1) we have

$$p_3(u;v) = \frac{v}{2}\; \frac{\frac{1}{2}\, v(1+u)}{\sigma_k^2}\; \exp\left\{-\frac{A^2+\left[\frac{1}{2}\, v(1+u)\right]^2}{2\sigma_k^2}\right\} I_0\left[\frac{A}{\sigma_k^2}\, \frac{1}{2}\, v(1+u)\right]$$

$$\cdot\; \frac{\frac{1}{2}\, v(1-u)}{\sigma_k^2}\; \exp\left\{-\frac{\left[\frac{1}{2}\, v(1-u)\right]^2}{2\sigma_k^2}\right\} \tag{5D-3}$$

and with algebraic manipulation of terms we obtain

$$p_3(u,v) = \frac{v^3(1-u^2)}{8\sigma_k^4} \exp\left(-\frac{A^2}{2\sigma_k^2}\right) \exp\left\{-\frac{v^2}{8\sigma_k^2}\left[(1+u)^2+(1-u)^2\right]\right\}$$

$$\cdot\; I_0\left[\frac{A\, v(1+u)}{2\sigma_k^2}\right]. \tag{5D-4}$$

The derived pdf is then obtainable by integrating out $p_3(u,v)$ with respect to v. Thus,

$$p_4(u) = \int_0^\infty dv\; p_3(u,v)$$

$$= \int_0^\infty dv\, \frac{v^3(1-u^2)}{8\sigma_k^4} \exp\left(-\frac{A^2}{2\sigma_k^2}\right) \exp\left\{-v^2\left[\frac{(1+u)^2+(1-u)^2}{8\sigma_k^4}\right]\right\}$$

$$\cdot\, I_0\left[v\,\frac{A(1+u)}{2\sigma_k^2}\right]$$

$$= \frac{(1-u^2)}{8\sigma_k^4} \exp\left(-\frac{A^2}{2\sigma_k^2}\right)\int_0^\infty v^3 \exp\left[-\left(\frac{1+u^2}{4\sigma_k^2}\right)v^2\right] I_0\left[v\,\frac{A(1+u)}{2\sigma_k^2}\right] dv. \tag{5D-5}$$

The last integral in (5D-5) may be evaluated using [2, eq. 6.631] to give

$$p_4(u) = \frac{(1-u^2)}{8\sigma^4}\; e^{-\rho}\; \frac{8\sigma^4}{(1+u^2)^2}\; {}_1F_1\left[2;\ 1;\ \frac{(1+u)^2}{1+u^2}\ \rho/2\right] \tag{5D-6}$$

where $\rho \triangleq A^2/2\sigma_k^2$. Making use of the fact that [18, eq. A.1.19c]

$${}_1F_1(2;\ 1;\ z) = (1+z)e^z, \tag{5D-7}$$

we can further simplify (5D-6). Thus,

$$p_4(u) = \frac{1-u^2}{(1+u^2)^2}\; e^{-\rho}\left[1+\rho\,\frac{(1+u)^2}{2(1+u^2)}\right]\exp\left\{\rho\left[\frac{(1+u)^2}{2(1+u^2)}\right]\right\} \tag{5D-8}$$

or, equivalently,

$$p_4(u) = \frac{(1-u^2)}{(1+u^2)^2}\left[1 + \rho\,\frac{(1+u)^2}{2(1+u^2)}\right]\exp\left\{-\rho\left[\frac{(1+u)^2}{2(1+u^2)}\right]\right\} \tag{5D-9}$$

which is the proof of equation (5-70).

As a check on the result, for the special case of L=1, the bit error probability is given by

$$P_b(e) = \int_{-1}^{0} du\,\frac{(1-u^2)}{(1+u^2)^2}\left[1 + \rho\,\frac{(1+u)^2}{2(1+u^2)}\right]\exp\left\{-\rho\left[\frac{(1+u)^2}{2(1+u^2)}\right]\right\}. \tag{5D-10}$$

Let $$x = \frac{(1-u)^2}{2(1+u^2)}\ ;$$

then

$$\frac{dx}{du} = \frac{-(1-u)(1+u)}{(1+u^2)^2} = -\frac{(1-u)^2}{(1+u^2)^2}. \tag{5D-11}$$

Using the transformation (5D-11) and recognizing that $(1+u)^2 = 2(1+u^2) - (1-u)^2$, we obtain

$$P_b(e) = \int_{1}^{1/2} -dx\left[1 + \rho(1-x)\right]e^{-\rho x}$$

$$= \int_{1/2}^{1} dx\left[1 + \rho(1-x)\right]e^{-\rho x} \tag{5D-12}$$

which gives

$$P_b(e) = \frac{1}{2}e^{-\rho/2}. \tag{5D-13}$$

Therefore, the result for L=1 is identical to the result obtained from the receiver using the square-law detector for the same L=1 hop/bit.

APPENDIX 5E

COMPUTER PROGRAM TO COMPUTE BIT ERROR PROBABILITY FOR LINEAR-LAW SELF-NORMALIZING RECEIVER WITH AN N-LEVEL QUANTIZER

The program for a linear-law self-normalizing receiver with an N-level quantizer is nearly identical to the program given in Appendix 5C for the square-law self-normalizing receiver with an N-level quantizer. The only change involved is replacing the subroutine VALUE in the program contained in Appendix 5C with the subroutine given on the following page. The user may also find it useful to modify the FORMAT statement in the subroutine PUT1 in Appendix 5C to provide identification of the results as pertaining to the linear-law receiver.

```
              SUBROUTINE VALUE(RHO,TRES,NQ,V)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                         C
      C COMPUTE DISCRETE PROBABILITIES FOR EACH QUANTIZED LEVEL C
      C             FOR LINEAR DETECTOR ONLY                    C
      C                                                         C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              DIMENSION V(NQ)
              DNQ=NQ
              DO 12 IQ=1,NQ
              V(IQ)=0.D0
      12      CONTINUE
      C     " for general N-level ( N not equal to 2) "
      C     -------------------------------------------
              IF (NQ.NE.2) THEN
                IF(TRES.LT.1.D0)THEN
                  UP=(1.D0+TRES)*(1.D0+TRES)/(2.D0*(1.D0+TRES*TRES))
                  DN=(1.D0-TRES)*(1.D0-TRES)/(2.D0*(1.D0+TRES*TRES))
                  V(1)=(1.D0-UP)*DEXP(-RHO*UP)
                  V(NQ)=1.D0-(1.D0-DN)*DEXP(-RHO*DN)
                  DO 103 IQ=2,NQ-1
                  DIQ=IQ
                  A9=DMIN1(1.D0,2.D0*TRES*(DIQ-DNQ/2.D0)/(DNQ-2.D0))
                  A8=DMAX1(-1.D0,2.D0*TRES*(DIQ-1.D0-DNQ/2.D0)/(DNQ-2.D0))
                  UPP=(1.D0-A9)*(1.D0-A9)/(2.D0*(1.D0+A9*A9))
                  DNN=(1.D0-A8)*(1.D0-A8)/(2.D0*(1.D0+A8*A8))
                  V(IQ)=(1.D0-UPP)*DEXP(-RHO*UPP)-(1.D0-DNN)*DEXP(-RHO*DNN)
      103         CONTINUE
                ELSE
                  DO 100 IQ=1,NQ
                  DIQ=IQ
                  A9=DMIN1(1.D0,2.D0*TRES*(DIQ-DNQ/2.D0)/(DNQ-2.D0))
                  A8=DMAX1(-1.D0,2.D0*TRES*(DIQ-1.D0-DNQ/2.D0)/(DNQ-2.D0))
                  A8NEXT=DMAX1(-1.D0,2.D0*TRES*(DIQ-DNQ/2.D0)/(DNQ-2.D0))
                  UPPP=(1.D0-A9)*(1.D0-A9)/(2.D0*(1.D0+A9*A9))
                  DNNN=(1.D0-A8)*(1.D0-A8)/(2.D0*(1.D0+A8*A8))
                  IF(A8.EQ.-1.D0.AND.A8NEXT.EQ.-1.D0) GOTO 100
                  V(IQ)=(1.D0-UPPP)*DEXP(-RHO*UPPP)
          $         -(1.D0-DNNN)*DEXP(-RHO*DNNN)
                  IF(A9.EQ.1.D0) GOTO 101
      100         CONTINUE
                ENDIF
      101       CONTINUE
      C     " for special case N=2 (threshold must be 0) "
      C     ----------------------------------------------
              ELSE IF(NQ.EQ.2 ) THEN
                V(1)=.5D0*DEXP(-RHO/2.D0)
                V(2)=.5D0*DEXP(-RHO/2.D0)*(2.D0*DEXP(RHO/2.D0)-1.D0)
              ENDIF
              RETURN
              END
```

APPENDIX 5F
AN ALTERNATE FORM FOR (5-66)

In the main text, the L-fold convolution of the discrete probability density function of the decision variable is given as a double summation, which we repeat here for easy reference:

$$V_k^{(L)} = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i+j-1 = k}}^{(L-1)(N-1)+1} V_i \, V_j^{(L-1)} \tag{5F-1}$$

The constraint $i + j - 1 = k$ in (5F-1) restricts the summation over j to a single term for each value of i, namely $j = k - i + 1$. Furthermore, the maximum value of k is readily found from the end points, $i = N$ and $j = (L-1) \cdot (N-1) + 1$, or

$$k_{max} = N + (L-1)(N-1) + 1 = L(N-1) + 1 \tag{5F-2}$$

Therefore, the L-fold self-convolution may also be written as the single summation

$$V_k^{(L)} = \sum_{i=1}^{N} V_i \, V_{k-i+1}^{(L-1)}, \quad k = 1, 2, \ldots, L(N-1) + 1. \tag{5F-3}$$

APPENDIX 8A

NUMBER OF WAYS α JAMMED HOPS MAY BE DISTRIBUTED OVER GROUPS OF L_1 AND L_2 HOPS WITHOUT REGARD TO ORDER WITHIN EACH GROUP

Out of a total of $L_1 + L_2$ hops, α_i hops are jammed with $0 \leq \alpha \leq L_1 + L_2$. The total number of hops is partitioned into two disjoint groups of L_1 and L_2 hops, respectively. We desire to count the number of ways the jammed hops can be split between the two groups without regard to the sequence of jammed hops in each group.

Without loss of generality, we may assume $L_1 \leq L_2$ (we may relabel the two groups, if necessary, to achieve this). If $\alpha \leq L_1$, then we may have:

0 jammed hops in L_1 hops, α jammed hops in L_2 hops;

or

1 jammed hop in L_1 hops, α-1 jammed hops in L_2 hops;

or

2 jammed hops in L_1hops, α-2 jammed hops in L_2 hops;

⋮

or α jammed hops in L_1, 0 jammed hops in L_2.

The total number of such ways is clearly α+1. On the other hand, if $\alpha > L_1$, then there can be no more than L_1 jammed hops in L_1 hops, and hence there are 0 or 1 or ... or L_1 jammed hops in L_1 hops and correspondingly α or α-1 or ... or α-L_1 jammed hops in L_2 hops, for a total of $L_1 + 1$ ways. We may then conclude there are $1 + \min(\alpha, L_1, L_2)$ ways of splitting the α jammed hops between the two groups without regard to the sequence of jammed hops in each group.

APPENDIX 8B

A NUMERICAL ALGORITHM FOR COMPUTING THE JAMMING EVENT PROBABILITIES $\pi_L(\underline{\ell})$

For general M, the analytical expression to compute the jamming event probabilities for L-hops/symbol becomes quite involved. A more practical approach is, therefore, needed. Since the differences between the tone jamming models are all reflected in the event probabilities on a per-hop basis, only one program is needed to compute the event probabilities for the general case of L hops/symbol. However, before going into the details of the program coding, we first show, by example, the algorithm that is needed.

Assume that on a given hop for, say M=2, Pr{0, 0} = a, Pr{0, 1} = b, Pr{1, 0} = c and Pr{1, 1} = d. We further assume that the jamming events on each one of the L hops are independent. For L=2, the jamming events with their corresponding event probabilities can be described by the matrix illustrated in Figure 8B-1. In Figure 8B-1a, the jamming events for the 2-hop case are obtained by adding the corresponding digits of the rows and columns of this array and recording the result at the corresponding intersection. Thus, at the intersection of (1,0) and (1,1), we obtain (2,1) and at the intersection of (1,1) and (1,1), we obtain (2,2). In Figure 8B-1b, the event probabilities for the 2-hop case are obtained by multiplying the event probabilities of the corresponding columns and rows of the array and recording the result at the corresponding intersection. By combining the two figures, we thus obtain the jamming events with their corresponding event probabilities for L=2 hops/symbol. However, if we carefully examine the matrix, we note that the event (1,1) appears four times and the events (0,1), (1,0), (1,2), and (2,1) each appear twice. The next step is, therefore, to combine the events that are the same

(a)

		EVENTS ON HOP 1			
		0 0	0 1	1 0	1 1
EVENTS ON HOP 2	0 0	0 0	0 1	1 0	1 1
	0 1	0 1	0 2	1 1	1 2
	1 0	1 0	1 1	2 0	2 1
	1 1	1 1	1 2	2 1	2 2

(b)

		EVENT PROBABILITIES ON HOP 1			
		a	b	c	d
EVENT PROBABILITIES ON HOP 2	a	a^2	ab	ac	ad
	b	ab	b^2	bc	bd
	c	ac	bc	c^2	cd
	d	ad	bd	cd	d^2

(c)

		HOP 1 PROBABILITY (EVENT)			
		a (0 0)	b (0 1)	c (1 0)	d (1 1)
HOP 2 PROBABILITY (EVENT)	a (0 0)	a^2 (0 0)	ab (0 1)	ac (1 0)	ad (1 1)
	b (0 1)	ab (0 1)	b^2 (0 2)	bc (1 1)	bd (1 2)
	c (1 0)	ac (1 0)	bc (1 1)	c^2 (2 0)	cd (2 1)
	d (1 1)	ad (1 1)	bd (1 2)	cd (2 1)	d^2 (2 2)

FIGURE 8B-1 RECURSIVE COMPUTATION OF JAMMING EVENT PROBABILITIES

and sum the probabilities of these events. This gives $\Pi_2(0,0) = a^2$, $\Pi_2(0,1) = 2ab$, $\Pi_2(0,2) = b^2$, $\Pi_2(1,0) = 2ac$, $\Pi_2(1,1) = 2ad + 2bc$, $\Pi_2(1,2) = 2bd$, $\Pi_2(2,0) = c^2$, $\Pi_2(2,1) = 2cd$, and $\Pi_2(2,2) = d^2$. If we sum all the event probabilities for the two hop case, we have

$$\sum_{\ell_1=0}^{2} \sum_{\ell_2=0}^{2} \Pi_2(\ell_1,\ell_2) = (a+b+c+d)^2. \tag{8B-1}$$

The jamming events and their corresponding event probabilities for the case of L=2 which are obtained by the above process can be used to determine the jamming events for L=3 and their corresponding probabilities by forming matrices with row elements equal to the events and probabilities for L=2 as just computed and column elements for L=1. We then repeat the process using this new array. In a similar fashion, we can iteratively compute event probabilities for any value of L.

The algorithm to generate the jamming events and their corresponding probabilities for the case of L hops can be summarized as follows.

1. Input values for L, M, N, q, and the jamming model, where

   L = number of hops/symbol,

   M = alphabet size,

   N = number of hopping frequencies,

   and q = number of jamming tones.

2. Using the above input values, compute $\Pr\{\ell_1, \ell_2, \ldots, \ell_M\}$ on a per-hop basis as a function of M, N, q, and the jamming model.
3. Discard those events for which the probability computed in step 2 is zero. If we were to omit this step, we would have $(L+1)^M$ distinct events to store in the computer's memory.

4. Consider the one-hop jamming event $\underline{\ell}$ as a subscript of an L-dimensional array of size 2 x 2 x...x 2 and map the subscripts of the non-zero jamming events into their equivalent linear subscripts according to the rule

$$\left.\begin{array}{l} \underbrace{(0,\ 0,\ 0,\ldots,0)}_{} = \text{ linear equivalent } 0 \\ (0,\ 0,\ 0,\ldots,1) = \text{ linear equivalent } 1 \\ \quad\vdots \\ \underbrace{(1,\ 1,\ 1,\ldots,1)}_{M \text{ dimensional}} = \text{ linear equivalent } 2^M-1 \end{array}\right\} NN$$

If all the elements have non-zero values, we will have mapped $NN = 2^M$ subscripts. If some events were discarded in step 3, we will have $NN < 2^M$.

5. Store the NN equivalent linear subscripts corresponding to the jamming events having non-zero probabilities compactly in an array, say ISUB(NN) such that

ISUB(1) = linear equivalent subscript of 1st event with non-zero probability

ISUB(2) = linear equivalent subscript of 2nd event with non-zero probability

⋮

ISUB(NN)= linear equivalent subscript of last event with non-zero probability

and store in another array, say A, the corresponding event probabilities,

A(1) = probability of 1st event with non-zero probability

A(2) = probability of 2nd event with non-zero probability

⋮

A(NN)= probability of last event with non-zero probability.

Thus A(i) contains the i^{th} non-zero event probability and ISUB(i) contains the equivalent linear subscript identifying the jamming event.

6. Copy array A into a second array, say D, and ISUB into a second array, say IDSUB.
7. Set LL=1.
8. Set arrays C and ICSUB to all zeros.
9. Use M nested loops, each running from 0 to 1, to index one-hop jamming events in array A and M additional nested loops, each running from 0 to LL, to index LL-hop jamming events in array D.
10. For the pair of events described by the loop indices, form the equivalent linear subscripts and search the arrays ISUB and IDSUB for their respective values. If either or both are not found, go to step 14. Else assume the equivalent linear subscripts were found at ISUB(i) and IDSUB(j), respectively.
11. Recover the vector subscripts corresponding to i and j, say I(1),...,I(M) and J(1),...,J(M), respectively, and form the subscript K(1),...,K(M) according to the rule K(m) = I(m) + J(m), m = 1,2,...,M.
12. Compute the equivalent linear subscript for K, assuming the array C to be (LL+1) x (LL+1) x...x (LL+1) elements. Search the ICSUB array for this entry. If found, record the location, say k, and go to step 13. Else make an entry in an available location in ICSUB and record this value as k.
13. Add the product A(i)· D(j) to C(k).
14. Step the nested loops begun in step 9. If all the loops are not exhausted, go to step 10.

15. If LL equals the desired number of hops L, go to step 19.
16. Copy arrays C and ICSUB into arrays D and IDSUB, respectively.
17. Increment LL by 1.
18. Go to step 8.
19. Sort the array ICSUB into ascending order, carrying the elements of the array C along with elements of array ICSUB as they are sorted.
20. For each element of array ICSUB, output the corresponding vector subscript as the jamming event and the corresponding value from array C as the probability of the jamming event.
21. Stop.

The reasons for linearizing the original M-dimensional vector subscript are related to computer memory and FORTRAN restrictions. With an M-dimensional vector subscript, say an array of the form

$$\text{DIMENSION A}(L+1,\ L+1,\ldots,L+1),$$

a total of $(L+1)^M$ numbers must be stored if all elements are stored. For L=4 and M=8, assuming 4 bytes of memory per floating point number stored, each array would require 4×5^8 = 1,562,500 bytes of memory. If the array is sparse, we can save substantial amounts of memory by storing only nonzero elements, along with the corresponding subscripts. Linearizing the subscripts facilitates storage of the subscript for arbitrary M without wasting memory on a multidimensional subscript array since FORTRAN must define the array's dimensionality at compile time. Linearizing the subscript also avoids limitations imposed by the maximum of 7 subscripts allowed by FORTRAN, which would otherwise restrict the program to $M \leq 7$.

Since the key to the algorithm is the linearization of the subscripts, an algorithm for this process is required. A suitable algorithm, based on the

lexicographical storage sequence of FORTRAN (i.e. first subscript varying most rapidly) is as follows. Define the array A by the pseudo-FORTRAN-77 statement

$$\text{DIMENSION } A(L_1{:}U_1, L_2{:}U_2, \ldots, L_M{:}U_M)$$

where the i-th subscript ranges from L_i to U_i. Then the offset from the array origin (in units equal to the size of one element of A) of element $A(\ell_1, \ell_2, \ldots, \ell_M)$ is

$$\text{LINEAR} = ((U_1-L_1+1)((\cdots(U_{M-2}-L_{M-2}+1)((U_{M-1}-L_{M-1}+1)(\ell_M-L_M) + (\ell_{M-1}-L_{M-1})) + \cdots) + (\ell_2-L_2))) + (\ell_1-L_1). \tag{8B-2}$$

The vector subscript may be recovered by the following algorithm:

1. Set TEMP = LINEAR
2. Set $I = 1$
3. Set $L = U_i - L_i + 1$.
4. Set ℓ_i = (TEMP mod L) + L_i.
5. Set TEMP = $\lfloor \text{TEMP}/L \rfloor$.
6. Set $I = I + 1$.
7. Repeat steps 3 through 6 while $I \le M$.
8. Stop.

APPENDIX 8C

COMPUTER PROGRAM TO COMPUTE JAMMING EVENT PROBABILITIES

The following pages contain a listing of a FORTRAN-77 computer program which implements the algorithm described in Appendix 8B. The subroutines INPUT and INPUT1, which were separately compiled, implement the 1-hop probability calculations for two different jamming models.

```
PDP-11 FORTRAN-77 V4.0-1        14:13:04    19-Jan-84          Page 1

      PROGRAM MODEL2
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                            C
C MODEL 2:  EVENLY SPACED TONES (CLUSTER OF SIZE 1)          C
C MODEL 3:  EVENLY SPACED CLUSTERED OF TONES FOR             C
C           SIZE .GE. 2 WITH SPACING GIVEN BY (M+1-n)        C
C                                                            C
C THIS PROGRAM COMPUTES JAMMING EVENTS PROBABILITIES FOR     C
C MULTITONE JAMMING MODEL (MODEL 2 AND 3) WHERE :            C
C                                                            C
C (A) FOR MODEL 2, THE JAMMING TONES ARE NOT CLUSTERED       C
C        AND ARE EVENLY SPACED IN FREQUENCY BY SOME          C
C        MULTIPLE (n) OF THE HOPPING RATE (SINGLE CLUSTER) C
C                                                            C
C (B) FOR MODEL 3, THE JAMMING TONES ARE CLUSTERED           C
C        WITH CLUSTER SIZE EQUAL TO n.                       C
C        THE SPACING BETWEEN ADJACENT CLUSTERS IS            C
C        GIVEN BY (M+1-n).                                   C
C                                                            C
C ASSUMING THE JAMMER KNOWS M, THIS STRATEGY ENABLES         C
C THE JAMMER TO CONTROL HOW MANY TONES JAM THE M-ARY         C
C SIGNAL BAND ON A GIVEN HOP, WHEN JAMMED.                   C
C                                                            C
C THE SIZE OF THE JAMMED PORTION OF THE BANDWIDTH CAN        C
C ALSO BE CONTROLLED WHEN THE NUMBER OF TONES IS FIXED.      C
C                                                            C
C PROGRAMMER: A. KADRICHU                                    C
C DATE: JAN 18, 1984                                         C
C                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT INTEGER*4(I-N),DOUBLE PRECISION (A-H,O-Z)
      LOGICAL*1 GO,GO2
      DIMENSION IASUB(500),ICSUB(500),IDSUB(500)
      DIMENSION LUP2(16),LUP3(16)
      DIMENSION KASUB(16),KCSUB(16),KDSUB(16)
      DIMENSION A(100),C(500),D(500)
      DIMENSION ILOWA(16),IUPA(16),ISUBA(16)
      DIMENSION ILOWC(16),IUPC(16),ISUBC(16)
      DIMENSION ILOWD(16),IUPD(16),ISUBD(16)
      DIMENSION I(16),II(16),III(16),IIII(16)
      DIMENSION LINC(16),LLOW(16),LUP1(16)
      DATA ILOWA/16*0/,IUPA/16*1/
      DATA ILOWC/16*0/,IUPC/16*6/
      DATA ILOWD/16*0/,IUPD/16*6/
      DATA LINC/16*1/,LLOW/16*0/,LUP1/16*1/
C
C INTERACTIVE INPUT
C
      WRITE(5,108)
      READ(5,111) MODEL
      WRITE(5,110)

PDP-11 FORTRAN-77 V4.0-1        14:13:04    19-Jan-84          Page 2

      READ(5,111) L
      WRITE(5,112)
      READ(5,111) M
      WRITE(5,113)
      READ(5,111) N
      I500=500
      IF(MODEL.EQ.2) THEN
        WRITE(5,114)
        READ(5,111) NS
        WRITE(5,115)
1999    READ(5,111) NQ
        DM=M
        DN=N
        DNS=NS
        Q=NQ
        TESTQ=(DN-1-DM)/DNS+1.DO
        IF(NQ.GT.TESTQ) GOTO 1999
      ELSEIF(MODEL.EQ.3) THEN
        WRITE(5,1114)
        READ(5,111) NS
        WRITE(5,1150)
1998    READ(5,111) NQ
        IF(MOD(NQ,NS).NE.0) GOTO 1998
      ENDIF
      DM=M
      DN=N
      DNS=NS
      Q=NQ
108   FORMAT(' ENTER MODEL NUMBER ',$)
110   FORMAT(' VALUE OF L IS ',$)
111   FORMAT(I5)
112   FORMAT(' VALUE OF M IS ',$)
113   FORMAT(' NUMBER OF FREQUENCY HOPPING CELLS IS ',$)
114   FORMAT(' SEPARATION OF JAMMING TONES IN UNIT B HZ ',$)
115   FORMAT(' VALUE OF Q IS ',$)
1114  FORMAT(' SIZE OF CLUSTER IS ',$)
1150  FORMAT(' VALUE OF Q (MUST BE DIVISIBLE BY THE SIZE
     $  OF CLUSTER) IS ',$)
      DN1=DN-1.DO
      DN3=DN-3.DO
      DN7=DN-7.DO
C JAMMING PATTERN W/NON-ZERO PROBABILITY ON PER-HOP BASIS
      NUSEA=0
      CALL VLINIT(I,LLOW,M)
90    CONTINUE
      DO 7 NN=1,M
7     ISUBA(NN)=I(NN)
      CALL LOCN(M,ILOWA,IUPA,ISUBA,ISUB)
      AIN=0.0D0
      IF(MODEL.EQ.2)CALL INPUT(I,M,NS,DN,Q,AIN)
      IF(MODEL.EQ.3)CALL INPUT1(I,M,NS,DN,Q,AIN)
```

```
C STORE ONLY THE NONZERO VALUES OF THF INPUT ARRAY OF A
        CALL PUTIN(AIN,A,IASUB,NUSEA,I500,ISUB,IERR)
        CALL VLITER(I,LLOW,LUP1,LINC,M,GO)
        IF(GO) GOTO 90
C WRITE OUT PHI1(L1,L2,...,LM) FOR SINGLE HOP CASE
        WRITE(6,9593) MODEL
9593    FORMAT('                MODEL ',I2,//)
        IF(MODEL.EQ.2) WRITE(6,9595)M,N,NQ,NS
9595    FORMAT(' JAMMING PATTERNS FOR M=',I1,' N=',I4,' Q=',I4,' FOR
     $ L=1 AND TONE SEPARATION =',I1,/)
        IF(MODEL.EQ.3) WRITE(6,6595)M,N,NQ,NS
6595    FORMAT(' JAMMING PATTERNS FOR M=',I1,' N=',I4,' Q=',I4,' FOR
     $ L=1 AND CLUSTER SIZE= ',I1,/)
        WRITE(6,9594)
9594    FORMAT(' JAMMING PATTERN')
        DO 30 I4=1,NUSEA
        CALL VECSUB(M,ILOWA,IUPA,IASUB(I4),KASUB)
        WRITE(6,900) (KASUB(ILOO),ILOO=1,M),A(I4)
900     FORMAT(' ',<M>I3,'            Pr(PATTERN)=',1PD10.3)
30      CONTINUE
C COMPUTATION STARTS HERE
        NUSED=0
        CALL VLINIT(I,LLOW,L)
99      CONTINUE
        DO 11 NN=1,M
11      ISUBA(NN)=I(NN)
        CALL LOCN(M,ILOWA,IUPA,ISUBA,ISUB)
        CALL LOOKUP(AOUT,A,IASUB,NUSEA,I500,ISUB)
        DIN=AOUT
        DO 16 NN=1,M
16      ISUBD(NN)=I(NN)
        CALL LOCN(M,ILOWD,IUPD,ISUBD,ISUB)
        CALL PUTIN(DIN,D,IDSUB,NUSED,I500,ISUB,IERR)
        CALL VLITER(I,LLOW,LUP1,LINC,M,GO)
        IF(GO) GOTO 99
C COMPUTATION OF PHI(L1,L2,L3,...,LM) FOR L HOPS/SYMBOL
        DO 9998 L1=1,L-1
9       DO 125 NN=1,M
        LUP2(NN)=L1
125     LUP3(NN)=L1+1
        NUSEC=0
        CALL VLINIT(I,LLOW,M)
98      CONTINUE
        CALL VLINIT(II,LLOW,M)
97      CONTINUE
        DO 18 NN=1,M
18      ISUBA(NN)=I(NN)
        CALL LOCN(M,ILOWA,IUPA,ISUBA,ISUB1)
        DO 19 NN=1,M
19      ISUBD(NN)=II(NN)
```



```
        CALL LOCN(M,ILOWD,IUPD,ISUBD,ISUB2)
        DO 21 NN=1,M
        III(NN)=I(NN)+II(NN)
        ISUBC(NN)=III(NN)
21      CONTINUE
        CALL LOCN(M,ILOWC,IUPC,ISUBC,ISUB3)
        CALL LOOKUP(AOUT,A,IASUB,NUSEA,I500,ISUB1)
        CALL LOOKUP(DOUT,D,IDSUB,NUSED,I500,ISUB2)
        CALL LOOKUP(COUT,C,ICSUB,NUSEC,I500,ISUB3)
        CIN=COUT+AOUT*DOUT
        CALL PUTIN(CIN,C,ICSUB,NUSEC,I500,ISUB3,IERR)
        CALL VLITER(II,LLOW,LUP2,LINC,M,GO2)
        IF(GO2) GOTO 97
        CALL VLITER(I,LLOW,LUP1,LINC,M,GO)
        IF(GO) GOTO 98
C
C STORE FOR NEXT ITERATIVE HOP COMPUTATIONS
C
        NUSED=0
        CALL VLINIT(IIII,LLOW,M)
96      CONTINUE
        DO 23 NN=1,M
23      ISUBC(NN)=IIII(NN)
        CALL LOCN(M,ILOWC,IUPC,ISUBC,ISUB)
        CALL LOOKUP(COUT,C,ICSUB,NUSEC,I500,ISUB)
        DIN=COUT
        DO 24 NN=1,M
24      ISUBD(NN)=IIII(NN)
        CALL LOCN(M,ILOWD,IUPD,ISUBD,ISUB)
        CALL PUTIN(DIN,D,IDSUB,NUSED,I500,ISUB,IERR)
        CALL VLITER(IIII,LLOW,LUP3,LINC,M,GO)
        IF(GO) GOTO 96
9998    CONTINUE
        WRITE(6,3333)M,L
3333    FORMAT(////' THE JAMMING EVENT PROBABILITY FOR M=',I2,' AND
     $ L=',I3,' HOPS/SYMBOL',/)
C
C WRITE OUT PHI(L1,L2,...,LM)
C
        WRITE(6,3434)
3434    FORMAT(' JAMMING PATTERN ')
        TOTAL=0.D0
        DO 3001 I4=1,NUSED
        CALL VECSUB(M,ILOWD,IUPD,IDSUB(I4),KDSUB)
        TOTAL=D(I4)+TOTAL
        WRITE(6,1001) (KDSUB(ILOO),ILOO=1,M),D(I4),TOTAL
1001    FORMAT(' ',<M>I5,'      PHI(PATTERN)=',1PD10.3,
     $  '    SUM=',0PF11.9)
3001    CONTINUE
        STOP
        END
```

PDP-11 FORTRAN-77 V4.0-1 14:13:34 19-Jan-84 Page 5

```
              SUBROUTINE PUTIN(CIN,C,ICSUB,NUSE,NMAX,K,IERR)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                              C
      C THIS SUBROUTINE INSERTS AN ELEMENT INTO A SPARSE ARRAY FOR    C
      C WHICH ONLY THE NON-ZERO ELEMENTS ARE KEPT IN STORAGE.         C
      C THE DOUBLE PRECISION VALUE CIN IS STORED AS C(K) WHERE        C
      C THE AUXILIARY ARRAY ICSUB IS USED TO KEEP TRACK OF THE        C
      C SUBSCRIPTS OF THE CORRESPONDING ENTRIES IN C.                 C
      C LONG (4-BYTE) INTEGERS ARE USED TO ACCOMODATE LARGE SUBSCRIPT C
      C VALUES FOR THE SPARSE ARRAY C                                 C
      C                                                              C
      C USAGE:                                                       C
      C       INTEGER*4 ICSUB(NMAX),NUSE,NMAX,K,IERR                 C
      C       DOUBLE PRECISION C(NMAX),CIN                           C
      C       CALL PUTIN(CIN,C,ICSUB,NUSE,NMAX,K,IERR)               C
      C WHERE                                                        C
      C  CIN = VALUE OF ELEMENT TO STORE                             C
      C  C = ARRAY IN WHICH NON-ZERO VALUES ARE ACTUALLY STORES      C
      C  ICSUB = AUXILIARY ARRAY FOR STORING ACTUAL SUBSCRIPT VALUES C
      C  NUSE = NUMBER OF ELEMENTS OF C CURRENTLY OCCUPIED           C
      C  NMAX = SIZE OF ARRAY C                                      C
      C  IERR = ERROR RETURN CODE, 0 IF NO ERROR OR 1 IF THERE IS    C
      C      NO ROOM IN C FOR THE INSERTION OF ANOTHER NON-ZERO VALUE C
      C NOTE: IF CIN=0 AND THE SUBSCRIPT K IS FOUND IN ICSUB, THEN    C
      C       THE ELEMENT IS DELETED BY SHIFTING DOWNWARD ALL        C
      C       FOLLOWING ELEMENTS OF THE ARRAY                        C
      C                                                              C
      C PROGRAMMER: ROBERT H. FRENCH          DATE: 11 JANUARY 1984   C
      C                                                              C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT INTEGER*4(I-N),DOUBLE PRECISION(A-H,O-Z)
              INTEGER*4 K,ICSUB(NMAX)
              DIMENSION C(NMAX)
              IERR=0
              IF(CIN.EQ.0.D0) GOTO30
              DO 10 I=1,NUSE
              IF(ICSUB(I).NE.K) GOTO 10
              C(I)=CIN
              ICSUB(I)=K
              RETURN
      10      CONTINUE
              IF(NUSE.LT.NMAX) GOTO 20
              IERR=1
              RETURN
      20      NUSE=NUSE+1
              ICSUB(NUSE)=K
              C(NUSE)=CIN
              RETURN
      30      DO 40 I=1,NUSE
              J=I
              IF(ICSUB(I).EQ.K) GOTO 50
      40      CONTINUE
              RETURN
```

PDP-11 FORTRAN-77 V4.0-1 14:13:34 19-Jan-84 Page 6

```
      C
      C REMOVE THE ZEROED ELEMENT
      C
      50      DO 60 I=J,NUSE-1
              ICSUB(I)=ICSUB(I+1)
              C(I)=C(I+1)
      60      CONTINUE
              NUSE=NUSE-1
              RETURN
              END
```

PDP-11 FORTRAN-77 V4.0-1 14:13:38 19-Jan-84 Page 7

```
              SUBROUTINE LOOKUP(COUT,C,ICSUB,N,NMAX,K)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                                  C
      C THIS SUBROUTINE RETRIEVES AN ELEMENT OF A SPARSE ARRAY WHICH HAS  C
      C BEEN STORED COMPACTLY BY STORING ONLY NON-ZERO ELEMENTS.          C
      C                                                                  C
      C THE ARRAY IS DOUBLE PRECISION AND ALL INTEGERS ARE 4-BYTE INTEGERS C
      C TO ALLOW FOR LARGE VALUES OF THE SUBSCRIPT OF THE SPARSE ARRAY.   C
      C                                                                  C
      C USAGE:                                                           C
      C       INTEGER*4 ICSUB(NMAX),N,NMAX,K                             C
      C       DOUBLE PRECISION C(NMAX),C                                 C
      C       CALL LOOKUP(COUT,C,ICSUB,N,NMAX,K)                         C
      C WHERE                                                            C
      C       COUT = VALUE OF C(K) (OUTPUT FROM SUBROUTINE)              C
      C          C = ARRAY USED TO STORE NON-ZERO ELEMENTS               C
      C      ICSUB = AUXILIARY ARRAY USED TO STORE ACTUAL SUBSCRIPTS     C
      C          N = NUMBER OF ELEMENTS OF C CURRENTLY IN USE            C
      C       NMAX = SIZE OF C                                           C
      C          K = SUBSCRIPT OF SPARSE ARRAY TO LOOK UP                C
      C                                                                  C
      C PROGRAMMER: ROBERT H. FRENCH                                     C
      C       DATE: 11 JANUARY 1984                                      C
      C                                                                  C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              IMPLICIT INTEGER*4(I-N),DOUBLE PRECISION(A-H,O-Z)
              INTEGER*4 ICSUB(NMAX),K
              DIMENSION C(NMAX)
              DO 10 I=1,N
              IF(ICSUB(I).NE.K)GOTO 10
              COUT=C(I)
              RETURN
      10      CONTINUE
              COUT=0.
              RETURN
              END
```

```
PDP-11 FORTRAN-77 V4.0-1        14:13:41     19-Jan-84           Page 8

0001            SUBROUTINE LOCN(NDIM,ILOW,IUP,ISUB,LINEAR)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                                   C
        C THIS SUBROUTINE COMPUTES THE EQUIVALENT LINEAR SUBSCRIPT FOR      C
        C A MULTIDIMENSIONAL ARRAY OF NDIM DIMENSIONS                       C
        C                                                                   C
        C IF THE ARRAY A IS DEFINED AS                                      C
        C       DIMENSION A(ILOW(1):IUP(1),...,ILOW(NDIM):IUP(NDIM))        C
        C AND ISUB(1),...,ISUB(NDIM) IS A SET OF SUBSCRIPTS FOR A, THEN     C
        C THIS SUBROUTINE RETURNS IN LINEAR THE OFFSET FROM THE ORIGIN OF   C
        C A TO THE ELEMENT A(ISUB(1),...,ISUB(NDIM)), ASSUMING THE FIRST    C
        C SUBSCRIPT VARIES MOST RAPIDLY.                                    C
        C                                                                   C
        C USAGE:                                                            C
        C       DIMENSION ILOW(NDIM),IUP(NDIM),ISUB(NDIM)                   C
        C       DATA ILOW/set lower limits of defined subscripts of array/  C
        C       DATA IUP/set upper limits of defined subscripts of array/   C
        C       ...SET ISUB TO DESIRED SEQUENCE OF SUBSCRIPTS...            C
        C       CALL LOCN(NDIM,ILOW,IUP,ISUB,LINEAR)                        C
        C WHERE                                                             C
        C       NDIM = NUMBER OF DIMENSIONS THE ARRAY HAS                   C
        C       ILOW = ARRAY OF LOWER SUBSCRIPT BOUNDS                      C
        C        IUP = ARRAY OF UPPER SUBSCRIPT BOUNDS                      C
        C       ISUB = ARRAY CONTAINING SUBSCRIPT FOR WHICH LOCATION IS     C
        C              TO BE COMPUTED                                       C
        C     LINEAR = RETURNED VALUE OF OFFSET INTO ARRAY IN MEMORY        C
        C                                                                   C
        C NOTE: ALL INTEGERS ARE 4-BYTE INTEGERS TO ALLOW FOR LARGE         C
        C       SUBSCRIPT VALUES                                            C
        C                                                                   C
        C PROGRAMMER: ROBERT H. FRENCH                                      C
        C       DATE: 11 JANUARY 1984                                       C
        C                                                                   C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT INTEGER*4(I-N)
0003            DIMENSION ILOW(NDIM),IUP(NDIM),ISUB(NDIM)
0004            LINEAR=0
0005            DO 10 I=1,NDIM-1
0006            J=NDIM-I+1
0007            LINEAR=(LINEAR+(ISUB(J)-ILOW(J)))*(IUP(J-1)-ILOW(J-1)+1)
0008    10      CONTINUE
0009            LINEAR=LINEAR+ISUB(1)-ILOW(1)
0010            RETURN
0011            END
```

```
PDP-11 FORTRAN-77 V4.0-1        14:13:44     19-Jan-84           Page 9

0001            SUBROUTINE VECSUB(NDIM,ILOW,IUP,LINEAR,ISUB)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                                   C
        C GIVEN A LINEAR OFFSET INTO AN ARRAY A DESCRIBED BY                C
        C       DIMENSION A(ILOW(1):IUP(1),...,ILOW(NDIM):IUP(NDIM))        C
        C RETURN THE SUBSCRIPTS FOR THE ELEMENT                             C
        C                                                                   C
        C USAGE:                                                            C
        C       DIMENSION ILOW(NDIM),IUP(NDIM),ISUB(NDIM)                   C
        C       .                                                           C
        C       . (SET UP ILOW AND IUP TO DESCRIBE THE ARRAY A)             C
        C       .                                                           C
        C       LINEAR = (DESIRED OFFSET)                                   C
        C       CALL VECSUB(NDIM,ILOW,IUP,LINEAR,ISUB)                      C
        C WHERE                                                             C
        C       NDIM = NUMBER OF DIMENSION TO ARRAY A                       C
        C       ILOW = ARRAY CONTAINING LOWER LIMITS FOR EACH SUBSCRIPT     C
        C        IUP = ARRAY CONTAINING UPPER LIMITS FOR EACH SUBSCRIPT     C
        C     LINEAR = LINEAR OFFSET INTO ARRAY IN MEMORY                   C
        C       ISUB = RETURNED VECTOR OF NDIM SUBSCRIPTS FOR THE ELEMENT   C
        C                                                                   C
        C NOTE: ALL INTEGERS ARE 4-BYTE QUANTITIES TO ALLOW FOR LARGE       C
        C       SUBSCRIPT VALUES                                            C
        C                                                                   C
        C PROGRAMMER: ROBERT H. FRENCH                                      C
        C       DATE: 11 JANUARY 1984                                       C
        C                                                                   C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT INTEGER*4(I-N)
0003            DIMENSION ILOW(NDIM),IUP(NDIM),ISUB(NDIM)
0004            ITEMP=LINEAR
0005            DO 10 I=1,NDIM-1
0006            L=IUP(I)-ILOW(I)+1
0007            JTEMP=ITEMP/L
0008            ISUB(I)=ITEMP-L*JTEMP+ILOW(I)
0009            ITEMP=JTEMP
0010    10      CONTINUE
0011            ISUB(NDIM)=JTEMP+ILOW(NDIM)
0012            RETURN
0013            END
```

```
0001          SUBROUTINE VLINIT(LVEC,LLOW,LMAX)
       CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       C THIS SUBROUTINE INITIALIZES A "VECTOR DO-LOOP" STRUCTURE   C
       C DEFINED BY THE FOLLOWING PSEUDO-FORTRAN CODE:              C
       C      DO 100 LVEC(1)=LLOW(1),LUP(1),LINC(1)                 C
       C      DO 100 LVEC(2)=LLOW(2),LUP(2),LINC(2)                 C
       C                     .                                      C
       C                     .                                      C
       C      DO 100 LVEC(LMAX)=LLOW(LMAX),LUP(LMAX),LINC(LMAX)     C
       C                     .                                      C
       C                     . (STATEMENTS IN RANGE OF LOOP)        C
       C 100  CONTINUE                                              C
       C                                                            C
       C THE COMPANION ROUTINE VLITER HANDLES THE LOOP CONTROL AT THE C
       C CONTINUE STATEMENT IN THE ABOVE STRUCTURE                  C
       C                                                            C
       C USAGE:                                                     C
       C      LOGICAL*1 GO                                          C
       C      DIMENSION LVEC(LMAX),LLOW(LMAX),LUP(LMAX),LINC(LMAX)  C
       C (INITIALIZE ARRAY LLOW TO STARTING VALUES OF THE NESTED LOOPS)C
       C (INITIALIZE ARRAY LUP TO STOPING VALUES OF THE NESTED LOOPS) C
       C (INITIALIZE ARRAY LINC TO INCREMENTS OF THE LOOPS)         C
       C      CALL VLINIT(LVEC,LLOW,LMAX)                           C
       C   100 CONTINUE                                             C
       C         .                                                  C
       C         . (STATEMENTS IN RANGE OF LOOPS)                   C
       C         .                                                  C
       C      CALL VLITER(LVEC,LLOW,LUP,LINC,LMAX,GO)               C
       C      IF(GO)GOTO 100                                        C
       C WHERE                                                      C
       C    LVEC = ARRAY FOR STORAGE OF LOOP INDICES. LVEC(1) IS THE C
       C           OUTER-MOST LOOP; LVEC(LMAX), THE INNER-MOST LOOP. C
       C    LLOW = ARRAY FOR STORAGE OF LOOP STARTING VALUES, IN SAME C
       C           SEQUENCE AS LVEC                                 C
       C    LUP = ARRAY FOR STORAGE OF LOOP FINAL VALUES, IN SAME   C
       C          SEQUENCE AS LVEC                                  C
       C    LINC = ARRAY FOR STORAGE OF LOOP INCREMENTS, IN SAME    C
       C           SEQUENCE AS LVEC                                 C
       C    LMAX = NUMBER OF LOOPS NESTED                           C
       C    GO = LOGICAL VARIABLE, .TRUE. IF JUMP BACK TO BEGINNING OF C
       C         STATEMENTS IN THE RANGE OF THE LOOP SHOULD OCCUR,  C
       C         .FALSE. OTHERWISE (I.E. OUTER-MOST LOOP TERMINATED) C
       C                                                            C
       C PROGRAMMER: ROBERT H. FRENCH          DATE: 11 JANUARY 1984 C
       CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002          IMPLICIT INTEGER*4(I-N)
0003          DIMENSION LVEC(LMAX),LLOW(LMAX)
0004          DO 1 N=1,LMAX
0005          LVEC(N)=LLOW(N)
0006   1      CONTINUE
0007          RETURN
0008          END
```



```
0001          SUBROUTINE VLITER(LVEC,LLOW,LUP,LINC,LMAX,GO)
       CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       C                                                              C
       C SUBROUTINE TO HANDLE THE LOOP ITERATION LOGIC FOR A "VECTOR DO-LOOP"C
       C                                                              C
       C SEE DETAILED COMMENTS IN SUBROUTINE VLINIT FOR USAGE AND     C
       C PARAMETER DEFINITIONS                                        C
       C                                                              C
       C PROGRAMMER: ROBERT H. FRENCH                                 C
       C      DATE: 11 JANUARY 1984                                   C
       C                                                              C
       CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002          IMPLICIT INTEGER*4(I-N)
0003          LOGICAL*1 GO
0004          DIMENSION LVEC(LMAX),LLOW(LMAX),LUP(LMAX),LINC(LMAX)
0005          GO=.TRUE.
0006          DO 100 NDX=1,LMAX
0007          NSUB=LMAX+1-NDX
0008          LVEC(NSUB)=LVEC(NSUB)+LINC(NSUB)
0009          IF((LINC(NSUB).GE.0.AND.LVEC(NSUB).LE.LUP(NSUB))
             $ .OR.(LINC(NSUB).LT.0.AND.LVEC(NSUB).GE.LUP(NSUB))) RETURN
0010          LVEC(NSUB)=LLOW(NSUB)
0011   100    CONTINUE
0012          GO=.FALSE.
0013          RETURN
0014          END
```

```
PDP-11 FORTRAN-77 V4.0-1        13:05:40    19-Jan-84         Page 1

0001          SUBROUTINE INPUT(I,M,NS,DN,Q,AIN)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                            C
        C THIS SUBROUTINE COMPUTES THE EVENT PROBABILITY FOR ALL      C
        C POSSIBLE JAMMING PATTERNS WITH NON-ZERO PROBABILITY FOR     C
        C L=1 HOP/SYMBOL FOR EVENLY SPACED TONES.                    C
        C                                                            C
        C THE DISTANCE BETWEEN TWO ADJACENT TONES IS EQUAL TO A      C
        C MULTIPLE (n) OF THE HOPPING RATE.                          C
        C                                                            C
        C PROGRAMMER: A. KADRICHU      DATE: JAN 19, 1984            C
        C                                                            C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002          IMPLICIT INTEGER*4(I-N),DOUBLE PRECISION (A-H,O-Z)
0003          DIMENSION I(16)
0004          DM=M
0005          DNS=NS
0006          DN1=DN-1.DO
0007          DN3=DN-3.DO
0008          DN7=DN-7.DO
        C
        C FOR M=2 WITH TONE SPACED AT 1 AND 2 B HZ APART
        C
0009          IF(M.EQ.2.AND.NS.EQ.1) THEN
0010            IF(I(1).EQ.0.AND.I(2).EQ.0) AIN=(DN-Q-2.DO)/DN1
0011            IF(I(1).EQ.0.AND.I(2).EQ.1) AIN=1.DO/DN1
0012            IF(I(1).EQ.1.AND.I(2).EQ.0) AIN=1.DO/DN1
0013            IF(I(1).EQ.1.AND.I(2).EQ.1) AIN=(Q-1.DO)/DN1
        C
0014          ELSE IF(M.EQ.2.AND.NS.EQ.2) THEN
0015            IF(I(1).EQ.0.AND.I(2).EQ.0) AIN=(DN-2.DO*Q-1.DO)/DN1
0016            IF(I(1).EQ.0.AND.I(2).EQ.1) AIN=Q/DN1
0017            IF(I(1).EQ.1.AND.I(2).EQ.0) AIN=Q/DN1
        C
        C FOR M=4 WITH TONE SPACED AT 1,2,3, AND 4 B HZ APART
        C
0018          ELSE IF(M.EQ.4.AND.NS.EQ.1) THEN
0019            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0)
     $          AIN=(DN-Q-6.DO)/DN3
0020            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1)
     $          AIN=1.DO/DN3
0021            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.1)
     $          AIN=1.DO/DN3
0022            IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.1)
     $          AIN=1.DO/DN3
0023            IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0)
     $          AIN=1.DO/DN3
0024            IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0)
     $          AIN=1.DO/DN3
0025            IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.0)
     $          AIN=1.DO/DN3
0026            IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.1)
     $          AIN=(Q-3.DO)/DN3
```

```
PDP-11 FORTRAN-77 V4.0-1        13:05:40    19-Jan-84         Page 2

        C
0027          ELSE IF(M.EQ.4.AND.NS.EQ.2) THEN
0028            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0)
     $          AIN=(DN-2.DO*Q-5.DO)/DN3
0029            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1)
     $          AIN=1.DO/DN3
0030            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0)
     $          AIN=1.DO/DN3
0031            IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0)
     $          AIN=1.DO/DN3
0032            IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.1)
     $          AIN=(Q-1.DO)/DN3
0033            IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0)
     $          AIN=1.DO/DN3
0034            IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0)
     $          AIN=(Q-1.DO)/DN3
        C
0035          ELSE IF(M.EQ.4.AND.NS.EQ.3) THEN
0036            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0)
     $          AIN=(DN-3.DO*Q-4.DO)/DN3
0037            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1)
     $          AIN=1.DO/DN3
0038            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0)
     $          AIN=Q/DN3
0039            IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0)
     $          AIN=Q/DN3
0040            IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0)
     $          AIN=1.DO/DN3
0041            IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1)
     $          AIN=(Q-1.DO)/DN3
        C
0042          ELSE IF(M.EQ.4.AND.NS.EQ.4) THEN
0043            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0)
     $          AIN=(DN-4.DO*Q-3.DO)/DN3
0044            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1)
     $          AIN=Q/DN3
0045            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0)
     $          AIN=Q/DN3
0046            IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0)
     $          AIN=Q/DN3
0047            IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0)
     $          AIN=Q/DN3
        C
        C FOR M=8 WITH TONE SPACED AT 1 UP TO 8 B HZ APART
        C
0048          ELSE IF(M.EQ.8.AND.NS.EQ.1) THEN
0049            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $          .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $          AIN=(DN-Q-14.DO)/DN7
0050            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $          .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
     $          AIN=1.DO/DN7
```

```
0051        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.1)
          $ AIN=1.DO/DN7
0052        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.1.AND.I(8).EQ.1)
          $ AIN=1.DO/DN7
0053        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.1.AND.I(6).EQ.1.AND.I(7).EQ.1.AND.I(8).EQ.1)
          $ AIN=1.DO/DN7
0054        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
          $ .AND.I(5).EQ.1.AND.I(6).EQ.1.AND.I(7).EQ.1.AND.I(8).EQ.1)
          $ AIN=1.DO/DN7
0055        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.1
          $ .AND.I(5).EQ.1.AND.I(6).EQ.1.AND.I(7).EQ.1.AND.I(8).EQ.1)
          $ AIN=1.DO/DN7
0056        IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.1
          $ .AND.I(5).EQ.1.AND.I(6).EQ.1.AND.I(7).EQ.1.AND.I(8).EQ.1)
          $ AIN=1.DO/DN7
0057        IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0058        IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0059        IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0060        IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.1
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0061        IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.1
          $ .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0062        IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.1
          $ .AND.I(5).EQ.1.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0063        IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.1
          $ .AND.I(5).EQ.1.AND.I(6).EQ.1.AND.I(7).EQ.1.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0064        IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.1
          $ .AND.I(5).EQ.1.AND.I(6).EQ.1.AND.I(7).EQ.1.AND.I(8).EQ.1)
          $ AIN=(Q-7.DO)/DN7
      C
0065       ELSE IF(M.EQ.8.AND.NS.EQ.2) THEN
0066        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=(DN-2.DO*Q-13.DO)/DN7
0067        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
          $ AIN=1.DO/DN7
0068        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
```



```
0069        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.1)
          $ AIN=1.DO/DN7
0070        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0071        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
          $ .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.1)
          $ AIN=1.DO/DN7
0072        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
          $ .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0073        IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0074        IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.1
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0075        IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.1
          $ .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0076        IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.1
          $ .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.1)
          $ AIN=(Q-3.DO)/DN7
0077        IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0078        IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0079        IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
          $ .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0080        IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
          $ .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
          $ AIN=(Q-3.DO)/DN7
      C
0081       ELSE IF(M.EQ.8.AND.NS.EQ.3) THEN
0082        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=(DN-3.DO*Q-12.DO)/DN7
0083        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
          $ AIN=1.DO/DN7
0084        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
0085        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $ .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $ AIN=1.DO/DN7
```

```
0086        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
     $      AIN=1.DO/DN7
0087        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0088        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0089        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=(Q-1.DO)/DN7
0090        IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0091        IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0092        IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
     $      AIN=(Q-2.DO)/DN7
0093        IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0094        IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0095        IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
     $      AIN=(Q-2.DO)/DN7
      C
0096      ELSE IF(M.EQ.8.AND.NS.EQ.4) THEN
0097        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=(DN-4.DO*Q-11.DO)/DN7
0098        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
     $      AIN=1.DO/DN7
0099        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0100        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0101        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0102        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0103        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
     $      AIN=(Q-1.DO)/DN7
```



```
0104        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0105        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
     $      AIN=(Q-1.DO)/DN7
0106        IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0107        IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=(Q-1.DO)/DN7
0108        IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0109        IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=(Q-1.DO)/DN7
      C
0110      ELSE IF(M.EQ.8.AND.NS.EQ.5) THEN
0111        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=(DN-5.DO*Q-10.DO)/DN7
0112        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
     $      AIN=1.DO/DN7
0113        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0114        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0115        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=Q/DN7
0116        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=Q/DN7
0117        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0118        IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
     $      AIN=(Q-1.DO)/DN7
0119        IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
0120        IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
     $      AIN=(Q-1.DO)/DN7
0121        IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
     $      .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
     $      AIN=1.DO/DN7
```

```
0122          IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=(Q-1.DO)/DN7
      C
0123        ELSE IF(M.EQ.8.AND.NS.EQ.6) THEN
0124          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=(DN-6.DO*Q-9.DO)/DN7
0125          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
          $   AIN=1.DO/DN7
0126          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
          $   AIN=1.DO/DN7
0127          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0128          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0129          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0130          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0131          IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=1.DO/DN7
0132          IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
          $   AIN=(Q-1.DO)/DN7
0133          IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=1.DO/DN7
0134          IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
          $   AIN=(Q-1.DO)/DN7
      C
0135        ELSE IF(M.EQ.8.AND.NS.EQ.7) THEN
C             IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=(DN-7.DO*Q-8.DO)/DN7
0137          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
          $   AIN=1.DO/DN7
0138          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0139          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
```



```
0140          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0141          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0142          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0143          IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0144          IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=1.DO/DN7
0145          IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
          $   AIN=(Q-1.DO)/DN7
      C
0146        ELSE IF(M.EQ.8.AND.NS.EQ.8) THEN
0147          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=(DN-8.DO*Q-7.DO)/DN7
0148          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
          $   AIN=Q/DN7
0149          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0150          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0151          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0152          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0153          IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0154          IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0155          IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $   .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $   AIN=Q/DN7
0156        ENDIF
0157        RETURN
0158        END
```

```
0001          SUBROUTINE INPUT1(I,M,NS,DN,Q,AIN)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C                                                          C
      C THIS SUBROUTINE COMPUTES THE EVENT PROBABILITIES OF ALL   C
      C POSSIBLE JAMMING PATTERNS WITH NON-ZERO PROBABILITY FOR   C
      C L=1 HOP/SYMBOL FOR EVENLY SPACED CLUSTER OF TONES WITH    C
      C SIZE n. THE DISTANCE BETWEEN TWO ADJACENT CLUSTERS        C
      C IS GIVEN BY (M+1-n).                                      C
      C                                                          C
      C THE TOTAL NUMBER OF TONES MUST BE EQUAL TO A MULTIPLE (n) C
      C OF THE CLUSTER SIZE.                                      C
      C                                                          C
      C PROGRAMMER: A. KADRICHU                                   C
      C DATE: JAN 19, 1984                                        C
      C                                                          C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002          IMPLICIT INTEGER*4(I-N),DOUBLE PRECISION (A-H,O-Z)
0003          DIMENSION I(16)
0004          DM=M
0005          DNS=NS
0006          DN1=DN-1.D0
0007          DN3=DN-3.D0
0008          DN7=DN-7.D0
      C
      C FOR M=2 WITH TONE CLUSTER OF SIZE 2
      C
0009          IF(M.EQ.2.AND.NS.EQ.2) THEN
0010            IF(I(1).EQ.0.AND.I(2).EQ.0) AIN=(DN-Q-2.D0)/DN1
0011            IF(I(1).EQ.0.AND.I(2).EQ.1) AIN=1.D0/DN1
0012            IF(I(1).EQ.1.AND.I(2).EQ.0) AIN=1.D0/DN1
0013            IF(I(1).EQ.1.AND.I(2).EQ.1) AIN=(Q-1.D0)/DN1
      C
      C FOR M=4 WITH TONE CLUSTER OF SIZE 2 AND 3
      C
0014          ELSE IF(M.EQ.4.AND.NS.EQ.2) THEN
0015            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0)
          $     AIN=(DN-2.D0*Q-4.D0)/DN3
0016            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.1)
          $     AIN=(Q/2.D0)/DN3
0017            IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.0)
          $     AIN=(Q/2.D0)/DN3
0018            IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0)
          $     AIN=(Q/2.D0)/DN3
0019            IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1)
          $     AIN=(Q/2.D0-1.D0)/DN3
0020            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1)
          $     AIN=1.D0/DN3
0021            IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0)
          $     AIN=1.D0/DN3
      C
0022          ELSE IF(M.EQ.4.AND.NS.EQ.3) THEN
0023            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0)
          $     AIN=(DN-4.D0*Q/3.D0-5.D0)/DN3
```



```
0024            IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.1)
          $     AIN=(Q/3.D0)/DN3
0025            IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.0)
          $     AIN=(Q/3.D0)/DN3
0026            IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.1)
          $     AIN=(Q/3.D0-1.D0)/DN3
0027            IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.1)
          $     AIN=(Q/3.D0-1.D0)/DN3
0028            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1)
          $     AIN=1.D0/DN3
0029            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.1)
          $     AIN=1.D0/DN3
0030            IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0)
          $     AIN=1.D0/DN3
0031            IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0)
          $     AIN=1.D0/DN3
      C
      C FOR M=8 WITH TONE CLUSTER OF SIZE 2 AND 3
      C
0032          ELSE IF(M.EQ.8.AND.NS.EQ.2) THEN
0033            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $     AIN=(DN-4.D0*Q-8.D0)/DN7
0034            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.1)
          $     AIN=(Q/2.D0)/DN7
0035            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $     .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.1.AND.I(8).EQ.0)
          $     AIN=(Q/2.D0)/DN7
0036            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $     .AND.I(5).EQ.1.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $     AIN=(Q/2.D0)/DN7
0037            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
          $     .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $     AIN=(Q/2.D0)/DN7
0038            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.1
          $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $     AIN=(Q/2.D0)/DN7
0039            IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.0
          $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $     AIN=(Q/2.D0)/DN7
0040            IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
          $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
          $     AIN=(Q/2.D0)/DN7
0041            IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
          $     AIN=(Q/2.D0-1.D0)/DN7
0042            IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
          $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
          $     AIN=1.D0/DN7
```

```
0043                IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
              $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
              $     AIN=1.DO/DN7
       C
0044              ELSE IF(M.EQ.8.AND.NS.EQ.3) THEN
0045                IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
              $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
              $     AIN=(DN-8.DO*Q/3.DO-9.DO)/DN7
0046                IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
              $     .AND.I(5).EQ.0.AND.I(6).EQ.1.AND.I(7).EQ.1.AND.I(8).EQ.1)
              $     AIN=(Q/3.DO)/DN7
0047                IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
              $     .AND.I(5).EQ.1.AND.I(6).EQ.1.AND.I(7).EQ.1.AND.I(8).EQ.0)
              $     AIN=(Q/3.DO)/DN7
0048                IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.1
              $     .AND.I(5).EQ.1.AND.I(6).EQ.1.AND.I(7).EQ.0.AND.I(8).EQ.0)
              $     AIN=(Q/3.DO)/DN7
0049                IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.1.AND.I(4).EQ.1
              $     .AND.I(5).EQ.1.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
              $     AIN=(Q/3.DO)/DN7
0050                IF(I(1).EQ.0.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.1
              $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
              $     AIN=(Q/3.DO)/DN7
0051                IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.1.AND.I(4).EQ.0
              $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
              $     AIN=(Q/3.DO)/DN7
0052                IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
              $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
              $     AIN=(Q/3.DO-1.DO)/DN7
0053                IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
              $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.1)
              $     AIN=(Q/3.DO-1.DO)/DN7
0054                IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
              $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.1)
              $     AIN=1.DO/DN7
0055                IF(I(1).EQ.0.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
              $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.1.AND.I(8).EQ.1)
              $     AIN=1.DO/DN7
0056                IF(I(1).EQ.1.AND.I(2).EQ.0.AND.I(3).EQ.0.AND.I(4).EQ.0
              $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
              $     AIN=1.DO/DN7
0057                IF(I(1).EQ.1.AND.I(2).EQ.1.AND.I(3).EQ.0.AND.I(4).EQ.0
              $     .AND.I(5).EQ.0.AND.I(6).EQ.0.AND.I(7).EQ.0.AND.I(8).EQ.0)
              $     AIN=1.DO/DN7
0058              ENDIF
0059              RETURN
0060              END
```

APPENDIX 8D

ALTERNATE FORMS FOR THE ONE-HOP JAMMING EVENT PROBABILITIES FOR INDEPENDENT MULTITONE JAMMING

The probability $\Pi_1(\underline{\nu})$ of the occurrence of the jamming event $\underline{\nu}$ is given by (8-35a)-(8-35c). These equations may also be expressed in a number of other forms. Beginning with the expression from (8-35c) we have the following progression of forms:

$$\Pi_1(\underline{\nu}) = \frac{(q-\ell+1)_\ell \; (N-q-M+1)_{M-\ell}}{(N-M+1)_M} \tag{8D-1}$$

$$= \frac{\Gamma(q+1)\Gamma(N-q+1-\ell)\Gamma(N-M+1)}{\Gamma(q-\ell+1)\Gamma(N-q-M+1)\Gamma(N+1)} \tag{8D-2}$$

$$= \frac{q!(N-q-\ell)!(N-M)!}{(q-\ell)!(N-q-M)!N!} \tag{8D-3}$$

$$= \frac{\binom{q}{\ell}\ell! \; (N-q-\ell)!}{\binom{N}{M} \; (N-q-M)!M!} \tag{8D-4}$$

$$= \frac{\binom{q}{\ell}\binom{N-q}{M} \; (N-q)!}{\binom{N}{M}\binom{N-q}{\ell} \; N!} \tag{8D-5}$$

$$= \frac{\binom{q}{\ell} \binom{N-q}{M}}{\binom{N}{M} \binom{N-q}{\ell} \binom{N}{q} \; q!} . \tag{8D-6}$$

APPENDIX 8E

DERIVATION OF APPROXIMATE FORMS FOR ERROR PROBABILITY OF BFSK/FH IN THE PRESENCE OF BOTH THERMAL NOISE AND TONE JAMMING

To obtain a more readily computed expression for the bit error probability, we approximate the noncentral χ^2 density function by a truncated Taylor series. We begin with the usual expression for the noncentral χ^2 density,

$$p_{\chi^2}(\alpha;\lambda,\nu) = \frac{1}{2}\exp\left(-\frac{\alpha+\lambda}{2}\right)\left(\frac{\alpha}{\lambda}\right)^{\frac{\nu-2}{4}} I_{\frac{\nu-2}{2}}(\sqrt{\alpha\lambda}), \tag{8E-1}$$

with noncentral parameter λ and ν degrees of freedom. If we replace the modified Bessel function in (8E-1) with its power series representation, we obtain

$$p_{\chi^2}(\alpha;\lambda,\nu) = \frac{1}{2}\exp\left(-\frac{\alpha+\lambda}{2}\right)\sum_{n=0}^{\infty}\frac{\left(\frac{\sqrt{\alpha\lambda}}{2}\right)^{2n+\frac{\nu-2}{2}}\left(\frac{\alpha}{\lambda}\right)^{\frac{\nu-2}{4}}}{n!\,\Gamma\left(n+\frac{\nu}{2}\right)} \tag{8E-2}$$

which may be rearranged to yield the form

$$p_{\chi^2}(\alpha;\lambda,\nu) = e^{-\lambda/2}\sum_{n=0}^{\infty}\frac{(\lambda/2)^n}{n!}\cdot e^{-\alpha/2}\cdot\frac{1}{2}\cdot\left(\frac{\alpha}{2}\right)^{n+\frac{\nu}{2}-1}\frac{1}{\Gamma(n+\nu/2)} \tag{8E-3}$$

from which it is apparent that the noncentral χ^2 density may be expressed as an infinite series of central (λ=0) χ^2 densities,

$$p_{\chi^2}(\alpha;\lambda,\nu) = e^{-\lambda/2}\sum_{n=0}^{\infty}\frac{(\lambda/2)^n}{n!}\,p_{\chi^2}(\alpha;\,0,\,\nu+2n). \tag{8E-4}$$

We now desire to express $p_{\chi^2}(\alpha;\lambda,\nu)$ in a Taylor series with respect to the noncentral parameter λ. From (8E-4) we obtain the derivatives

$$\frac{\partial p_{\chi^2}}{\partial \lambda} = -\frac{1}{2}\, p_{\chi^2}(\alpha;\lambda,\nu) + \frac{1}{2}\, p_{\chi^2}(\alpha;\lambda,\nu+2) \tag{8E-5}$$

and

$$\frac{\partial^2 p_{\chi^2}}{\partial \lambda^2} = -\frac{1}{2}\left[-\frac{1}{2}\, p_{\chi^2}(\alpha;\lambda,\nu) + \frac{1}{2}\, p_{\chi^2}(\alpha;\lambda,\nu+2)\right]$$

$$+\frac{1}{2}\left[-\frac{1}{2}\, p_{\chi^2}(\alpha;\lambda,\nu+2) + \frac{1}{2}\, p_{\chi^2}(\alpha;\lambda,\nu+4)\right]$$

$$= \frac{1}{4}\left[p_{\chi^2}(\alpha;\lambda,\nu) - 2p_{\chi^2}(\alpha;\lambda,\nu+2) + p_{\chi^2}(\alpha;\lambda,\nu+4)\right]. \tag{8E-6}$$

If we set

$$\lambda = a + b\eta \tag{8E-7}$$

and write the Taylor series for $p_{\chi^2}(\alpha;\, a+b\eta,\nu)$ about the point $\lambda_0 = a$, we obtain the result

$$p_{\chi^2}(\alpha;\, a+b\eta,\nu) \simeq p_{\chi^2}(\alpha;a,\nu) + \frac{b\eta}{2}\left[p_{\chi^2}(\alpha;a,\nu+2) - p_{\chi^2}(\alpha;\, a,\nu)\right]$$

$$+\frac{1}{2}\left(\frac{b\eta}{2}\right)^2\left[p_{\chi^2}(\alpha;a,\nu) - 2p_{\chi^2}(\alpha;\, a,\nu+2) + p_{\chi^2}(\alpha;\, a,\nu+4)\right]. \tag{8E-8}$$

To apply the result (8E-8) to the problem of tone jamming, we refer to (8-7) for the noncentral parameter of the density function of the signal channel. If we let

$$\eta = \frac{1}{\ell_1} \sum_{\ell=1}^{\ell_1} \cos\theta_i = \frac{\zeta}{\ell_1} \tag{8E-9}$$

where the θ_i are the phase differences between the signal and the jamming tones on the ℓ_1 jammed hops, then we can use

$$E(\eta) = 0 \tag{8E-10}$$

and

$$E(\eta^2) = \frac{1}{\ell_1} \sum_{i=1}^{\ell_1} E\left(\cos^2\theta_i\right) = 1/2 \tag{8E-11}$$

in conjunction with (8E-8) to write

$$E_\eta\left\{p_{\chi^2}(\alpha;a+b\eta,\nu)\right\}$$

$$\simeq p_{\chi^2}(\alpha;a,\nu) + \frac{1}{2}\left(\frac{b}{2}\right)^2 \cdot \frac{1}{2}\left[p_{\chi^2}(\alpha;a,\nu) - 2p_{\chi^2}(\alpha;\ a,\nu+2) + p_{\chi^2}(\alpha;\ a,\nu+4)\right], \tag{8E-12}$$

or

$$p_{z_1}(\alpha|\ell_1) \simeq \left(1 + \frac{b^2}{16}\right) p_{\chi^2}(\alpha;a,\nu) + \frac{b^2}{16}\left[p_{\chi^2}(\alpha;a,\nu+4) - p_{\chi^2}(\alpha;\ a,\nu+2)\right] \tag{8E-13}$$

where

$$\nu = 2L, \tag{8E-14a}$$

$$a = \frac{2E_b}{N_0}\left(K + \frac{\ell_1}{\gamma\ E_b/N_J}\right), \tag{8E-14b}$$

and

$$b = 4\ell_1 \sqrt{\frac{K}{L} \cdot \frac{1}{\gamma E_b/N_J}} \quad \frac{E_b}{N_0}\ . \tag{8E-14c}$$

Upon substituting the explicit form for $p_{\chi^2}(\alpha;a,\nu)$ into (8E-13) and factoring out common terms, we obtain

$$p_{z_1}(\alpha|\ell_1) \approx \frac{1}{2}\exp\left(-\frac{\alpha+a}{2}\right)\left(\frac{\alpha}{a}\right)^{\frac{L-1}{2}}\left\{\left(1+\frac{b^2}{16}\right) I_{L-1}\left(\sqrt{\alpha a}\right)\right.$$

$$\left. + \frac{b^2}{16}\sqrt{\frac{\alpha}{a}}\left[\sqrt{\frac{\alpha}{a}}\; I_{L+1}\left(\sqrt{\alpha a}\right) - 2I_L\left(\sqrt{\alpha a}\right)\right]\right\}. \tag{8E-15}$$

Using the recurrence relation for the modified Bessel functions [4, eq. 9.6.26] in (8E-15), we can reduce the number of Bessel functions which must be computed by writing $I_{L+1}(\cdot)$ in terms of $I_L(\cdot)$ and $I_{L-1}(\cdot)$, with the result

$$p_{z_1}(\alpha|\ell_1) \approx \frac{1}{2}\exp\left(-\frac{\alpha+a}{2}\right)\left\{\left[1+\frac{b^2}{16}\left(1+\frac{\alpha}{a}\right)\right]\left(\frac{\alpha}{a}\right)^{\frac{L-1}{2}} I_{L-1}\left(\sqrt{\alpha a}\right)\right.$$

$$\left. - \frac{b^2}{8}\left(1+\frac{L}{a}\right)\left(\frac{\alpha}{a}\right)^{L/2} I_L\left(\sqrt{\alpha a}\right)\right\}. \tag{8E-16}$$

Finally, we use (8E-16) in (8-18) to obtain the approximate form for the conditional error probability,

$$P_s(e|\ell_1,\ell_2) = \int_0^\infty d\alpha\; p_{z_1}(\alpha|\ell_1)\int_0^\alpha d\beta\; p_{z_2}(\beta|\ell_2) \tag{8E-17a}$$

where the inner integral is readily computed in terms of the generalized Q-function [25] and $p_{z_1}(\alpha|\ell_1)$ is given by (8E-16) and (8E-14).

We further note that a one-term approximation to $p_{z_1}(\alpha|\ell_1)$, corresponding to taking only the first term of the Taylor series in (8E-18), may be obtained from (8E-16) by setting $b=0$ to give

$$p_{z_1}(\alpha|\ell_1) \approx \frac{1}{2}\exp\left(-\frac{\alpha+a}{2}\right)\left(\frac{\alpha}{a}\right)^{(L-1)/2} I_{L-1}\left(\sqrt{\alpha a}\right). \qquad \text{(8E-17b)}$$

The form given in (8E-17) may also be viewed as an approximation obtained by over-bounding $\cos\theta_i$ by one.

APPENDIX 8F

COMPUTER PROGRAM TO COMPUTE BIT ERROR PROBABILITY FOR BFSK/FH WITH L HOPS/BIT IN THE PRESENCE OF BOTH THERMAL NOISE AND INDEPENDENT MULTITONE JAMMING USING APPROXIMATE FORMULATION

The following pages contain a listing of the FORTRAN-77 program used to obtain numerical values of the probability of bit error for BFSK/FH with L hops/bit in the presence of both thermal noise and independent multitone (randomly placed tones) jamming using an approximate form for the signal-channel density function.

The function DXI used in this program is given in Appendix 4I. A listing of the numerical integration routine DGAU20 may be found in Appendix 4G, listing page 11, under the name DGAU. The function PNXY, which computes the generalized Q function using Shnidman's algorithm [25], is given in Appendix 8I. For subroutine DXBESI, see Appendix 4G, listing pages 12-13.

The constant PISQ in the function PDFZET is π^2.

```
          PROGRAM TJBLH
C
C COMPUTE THE ERROR PROBABILITY FOR BINARY FSK/FH WITH MULTIPLE
C HOPS/BIT IN THE PRESENCE OF INDEPENDENT MULTITONE JAMMING
C
C ANALYSIS: L. E. MILLER     PROGRAM: R. H. FRENCH
C
C V 4.2.0
C 26 OCT 1983, 3 NOV 83, 8 NOV 83, 25 JAN 84, 26 JAN 84
C
C 1- AND 3-TERM SERIES APPROXIMATIONS IN LIEU OF AVERAGING OVER
C THE DENSITY OF THE PHASE BETWEEN THE SIGNAL AND THE JAMMER.
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          COMMON /INPUTS/ DEBNOL(5),DEBNJL(16),LLIST(4),NSLOTS,
     $      GAMLST(31)
          COMMON /SIZE/ NO,NJ,NL,NG
          CALL GET
          SLOTS=NSLOTS
          PFAC=SLOTS/(SLOTS-1.DO)
          RSLOTS=1.DO/SLOTS
          DO 900 IL=1,NL
          LL=LLIST(IL)
          LSAVE=LL
          FLL=LL
          DO 800 IO=1,NO
          EBNO=10.DO**(DEBNOL(IO)/10.DO)
          SNR=EBNO/FLL
          DO 700 IG=1,NG
          GAMMA=GAMLST(IG)
          OMG=1.DO-GAMMA
          CALL PUT1(LL,NSLOTS,DEBNOL(IO),GAMMA)
          TCON1=10.DO/(GAMMA*FLL)
          TCON2=TCON1/100.DO
          DO 600 IJ=1,NJ
          R=10.DO**(DEBNJL(IJ)/10.DO)
          RJN=EBNO/(GAMMA*R)
          AJNR=10.DO*DLOG10(RJN)
          GMR=GAMMA-RSLOTS
C CALCULATE ELEMENTAL EVENT PROBABILITIES
          POO=OMG*(1.DO-GAMMA*PFAC)
          PO1=GAMMA*(1.DO-GMR*PFAC)
          P10=PO1
          P11=GAMMA*GMR*PFAC
C EVALUATE THE PROBABILITY
          CALL PSUBE(SNR,RJN,LL,POO,PO1,P10,P11,PE,PE2)
          CALL PUT2(DEBNJL(IJ),PE,PE2)
600       CONTINUE
700       CONTINUE
800       CONTINUE
900       CONTINUE
          STOP 0
          END
```



```
          SUBROUTINE GET
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          DIMENSION DO(5),DJ(16),DG(31)
          COMMON /INPUTS/ DEBNOL(5),DEBNJL(16),LLIST(4),NSLOTS,
     $      GAMLST(31)
          COMMON /SIZE/ NO,NJ,NL,NG
          DATA DO / 13.3525DO, 12.3133DO, 10.9444DO, 0.DO, 0.DO/
          DATA DJ/0.DO, 2.DO, 4.DO, 6.DO, 8.DO,10.DO,12.DO,14.DO,
     $    16.DO, 18.DO,20.DO,24.DO,28.DO,32.DO,40.DO,50.DO/
          DATA DG / .001DO, .01DO, .1DO, .5DO, 1.DO, 26*0.DO/
1         WRITE(5,2)
2         FORMAT(' HOW MANY VALUES OF EB/NO? [3] ',$)
          READ(5,3,ERR=1)NO
3         FORMAT(I1)
          IF(NO.EQ.0)NO=3
          IF(NO.LE.0.OR.NO.GT.5)GOTO 1
          DO 7 IN=1,NO
4         WRITE(5,5)IN,DO(IN)
5         FORMAT('   ENTER EB/NO(',I1,') [',F7.4,']: ',$)
          READ(5,6,ERR=4)DEBNOL(IN)
6         FORMAT(F7.4)
          IF(DEBNOL(IN).EQ.0.DO)DEBNOL(IN)=DO(IN)
7         CONTINUE
8         WRITE(5,9)
9         FORMAT(' HOW MANY VALUES OF EB/NJ? [16] ',$)
          READ(5,10,ERR=8)NJ
10        FORMAT(I2)
          IF(NJ.EQ.0)NJ=16
          IF(NJ.LE.0.OR.NJ.GT.16)GOTO 8
          DO 14 IN=1,NJ
11        WRITE(5,12)IN,DJ(IN)
12        FORMAT('   ENTER EB/NJ(',I2,') [',F4.1,']: ',$)
          READ(5,13,ERR=11)DEBNJL(IN)
13        FORMAT(F4.1)
          IF(DEBNJL(IN).EQ.0.DO)DEBNJL(IN)=DJ(IN)
14        CONTINUE
15        WRITE(5,16)
16        FORMAT(' HOW MANY VALUES OF L? [4] ',$)
          READ(5,17,ERR=14)NL
17        FORMAT(I1)
          IF(NL.EQ.0)NL=4
          IF(NL.LE.0.OR.NL.GT.4)GOTO 15
          DO 21 IN=1,NL
18        WRITE(5,19)IN,IN
19        FORMAT('   ENTER L(',I1,') [',I1,']: ',$)
          READ(5,20,ERR=18)LLIST(IN)
20        FORMAT(I2)
          IF(LLIST(IN).EQ.0)LLIST(IN)=IN
          IF(LLIST(IN).LE.0)GOTO 18
21        CONTINUE
22        WRITE(5,23)
23        FORMAT(' HOW MANY HOPPING SLOTS? [1000]: ',$)
```

PDP-11 FORTRAN-77 V4.0-1 13:26:07 8-May-84 Page 3

```
              READ(5,24,ERR=22)NSLOTS
        24    FORMAT(I5)
              IF(NSLOTS.EQ.0)NSLOTS=1000
              IF(NSLOTS.LE.1)GOTO 22
        25    WRITE(5,26)
        26    FORMAT(' HOW MANY VALUES OF GAMMA? [5]: ',$)
              READ(5,27,ERR=25)NG
        27    FORMAT(I2)
              IF(NG.EQ.0)NG=5
              IF(NG.LE.0.OR.NG.GT.31)GOTO 25
              DO 31 IN=1,NG
        28    WRITE(5,29)IN,DG(IN)
        29    FORMAT('    ENTER GAMMA(',I2,') [',1PD8.1,']: ',$)
              READ(5,30,ERR=28)GAMLST(IN)
        30    FORMAT(D15.8)
              IF(GAMLST(IN).EQ.0.D0)GAMLST(IN)=DG(IN)
              IF(GAMLST(IN).LE.0.D0.OR.GAMLST(IN).GT.1.D0)GOTO 28
        31    CONTINUE
              RETURN
              END
```

PDP-11 FORTRAN-77 V4.0-1 13:26:21 8-May-84 Page 4

```
              SUBROUTINE PUT1(L,NSLOTS,DEBNO,GAMMA)
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              WRITE(6,1)L,NSLOTS,DEBNO,GAMMA
        1     FORMAT('1BINARY FSK/FH (APPROXIMATE ANALYSIS)'/
           $ ' L = ',I1,' HOPS/BIT WITH ',I5,
           $ ' HOPPING SLOTS AVAILABLE'/
           $ ' EB/NO = ',F8.4,' DB'/' GAMMA = ',1PD10.3//
           $ 11X,'1-TERM',6X,'2-TERM'/
           $ '  EB/NJ',5X,'P(E)',8X,'P(E)')
              RETURN
              END
```

PDP-11 FORTRAN-77 V4.0-1 13:26:23 8-May-84 Page 5

```
              SUBROUTINE PUT2(DEBNJ,PE,PE2)
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              WRITE(6,1)DEBNJ,PE,PE2
        1     FORMAT(1X,F6.2,2X,1PD10.3,2X,1PD10.3)
              RETURN
              END
```

PDP-11 FORTRAN-77 V4.0-1 13:26:24 8-May-84 Page 6

```
              SUBROUTINE PSUBE(SNR,RJN,LL,P00,P01,P10,P11,PE,PE2)
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              LOGICAL*1 SV(20,105,0:4)
              COMMON /VALIDS/ SV
              PE=0.D0
              PE2=0.D0
              DO 1 LC=0,4
              DO 1 LB=1,105
              DO 1 LA=1,20
        1     SV(LA,LB,LC)=.FALSE.
              DO 100 L1=0,LL
              DO 100 L2=0,LL
              CALL FOOT(LL,L1,L2,P00,P01,P10,P11,PIE)
              IF(PIE.EQ.0.D0)GOTO 100
              CALL PELL(SNR,RJN,LL,L1,L2,PALL,PALL2)
              UEP=PIE*PALL
              UEP2=PIE*PALL2
              PE=PE+UEP
              PE2=PE2+UEP2
        100   CONTINUE
              RETURN
              END
```

PDP-11 FORTRAN-77 V4.0-1 13:26:29 8-May-84 Page 7

```
              SUBROUTINE FOOT(LL,L1,L2,P00,P01,P10,P11,PIE)
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              LUP=MIN0(L1,L2)
              LOW=MAX0(0,L1+L2-LL)
              SUM=0.D0
              DO 100 K=LOW,LUP
              PART=DMNC4(LL,K,L1-K,L2-K,LL-L1-L2+K)*DXI(P11,K)
           $   *DXI(P10,L1-K)*DXI(P01,L2-K)*DXI(P00,LL-L1-L2+K)
              SUM=SUM+PART
        100   CONTINUE
              PIE=SUM
              RETURN
              END
```

```
      DOUBLE PRECISION FUNCTION DMNC4(L,K1,K2,K3,K4)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DMNC4=0.D0
      IF(K1+K2+K3+K4.NE.L)RETURN
      IF(K1.LT.0.OR.K1.GT.L.OR.K2.LT.0.OR.K2.GT.L.OR.K3.LT.0
     $ .OR.K3.GT.L.OR.K4.LT.0.OR.K4.GT.L) RETURN
      DMNC4=1.D0
      DO 1 I=1,L
1     DMNC4=DMNC4*I
      IF(K1.NE.0)THEN
        DO 2 I=1,K1
2       DMNC4=DMNC4/I
      END IF
      IF(K2.NE.0)THEN
        DO 3 I=1,K2
3       DMNC4=DMNC4/I
      END IF
      IF(K3.NE.0)THEN
        DO 4 I=1,K3
4       DMNC4=DMNC4/I
      END IF
      IF(K4.NE.0)THEN
        DO 5 I=1,K4
5       DMNC4=DMNC4/I
      END IF
      RETURN
      END
```



```
      SUBROUTINE PELL(SNR,RJN,LL,L1,L2,PALL,PALL2)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      LOGICAL*1 NOSAVE,QUFL,AUFL,LEFT,RIGHT
      EXTERNAL GRAND,SECOND
      COMMON /VDX/ INTERV,LSH,INVOKE
      COMMON /IGP/ PLAM1,SHNID1,LPASS,LPM1,BSO16,NOSAVE
      COMMON /CONVRG/ QUFL,AUFL
      FLL=LL
      FL1=L1
      FL2=L2
      LSH=L2
      PLAM1=2.D0*(FLL*SNR+FL1*RJN)
      BSO16=SNR*FL1*RJN
      CONSTA=PLAM1
      NOSAVE=.FALSE.
      XINC=20.D0
      IF(CONSTA.GE.1000.D0) THEN
        NOSAVE=.TRUE.
        XINC=200.D0
        IF(CONSTA.GT.10000.D0) XINC=2000.D0
        IF(CONSTA.GT.1.D5) XINC=2.D4
        IF(CONSTA.GT.1.D6) XINC=2.D5
      END IF
302   KONSTA=CONSTA/XINC+0.5D0
      SHNID1=FL2*RJN/FLL
      LPASS=LL
      LPM1=LPASS-1
      PALL=0.D0
      XL=0.D0
      INTERV=1
      IF(KONSTA.GT.105) THEN
C WORK FROM THE MIDDLE OUT FOR LARGE NONCENTRAL PARAMETERS;
C SO WE CAN'T USE THE SAVED Q-FUNCTION ARRAY...
        NOSAVE=.TRUE.
        RIGHT=.FALSE.
        LEFT=.FALSE.
        INTERV=KONSTA
        LENGTH=INTERV
        XM=XINC*KONSTA
111     XU=XM
        XL=XU-XINC
        INVOKE=0
        CALL DGAU20(XL,XU,GRAND,ANSWER)
C IF ANSWER IS ZERO, WE AREN'T IN THE REGION WE WANT TO BE IN
        IF(ANSWER.EQ.0.D0) THEN
C IF 0 IS DUE TO Q FUNCTION UNDERFLOW, TRY FURTHER LEFT UNLESS
C LAST MOVE WAS RIGHT IN WHICH CASE STOP WITH 0 RESULT
          IF(QUFL.AND.(.NOT.AUFL)) THEN
C IF PREVIOUS STEP WAS RIGHT, BACK UP ONLY HALF WAY
            IF(.NOT.LEFT)THEN
              LENGTH=LENGTH/2
              IF(LENGTH.EQ.0) GOTO 800
            END IF
```

```
                LEFT=.TRUE.
                RIGHT=.FALSE.
                INTERV=INTERV-LENGTH
C IF WE CAN'T GO FURTHER LEFT, RETURN ZERO ANSWER
                IF(INTERV.LE.0) GOTO 800
                XM=INTERV*XINC
                GOTO 111
C IF 0 IS DUE TO PDF UNDERFLOW, TRY FURTHER RIGHT UNLESS
C LAST MOVE WAS LEFT IN WHICH CASE STOP WITH  0 RESULT
              ELSE IF((.NOT.QUFL).AND.AUFL) THEN
                IF(.NOT.RIGHT) THEN
                  LENGTH=LENGTH/2
                  IF(LENGTH.EQ.0) GOTO 800
                END IF
                RIGHT=.TRUE.
                LEFT=.FALSE.
                INTERV=INTERV+LENGTH
                XM=XINC*INTERV
                GOTO 111
C BOTH UNDERFLOW, RETURN 0 ANSWER
              ELSE IF(AUFL.AND.QUFL) THEN
                GOTO 800
              ELSE
C ONLY PRODUCT UNDERFLOWED, SO STARTING POINT IS OK
                GOTO 700
              END IF
C RETURN A ZERO ANSWER
800        PALL=0.D0
           RETURN
           END IF
C NOW WE HAVE A REGION WHERE SOMETHING NONZERO MAY ARISE, SO
C DO THE INTEGRAL BOTH WAYS FROM HERE. LEFTWARD FIRST.
700        PALL=ANSWER
           JNTERV=INTERV
           XU=XL
701        XL=XU-XINC
           INTERV=INTERV-1
C STOP AT THE ORIGIN
           IF(XL.LT.0.D0) GOTO 600
           INVOKE=0
           CALL DGAU20(XL,XU,GRAND,ANSWER)
           PALL=PALL+ANSWER
C IF WE REACH ZERO TAIL, GO ON TO THE RIGHTWARD PART
           IF(ANSWER.EQ.0.D0) GOTO 600
           GOTO 701
C NOW SET UP TO LET THE NORMAL CODE TAKE OVER FOR RIGHTWARD PART
600        XL=XM
           INTERV=JNTERV+1
         END IF
```



```
10         XU=XL+XINC
           INVOKE=0
           CALL DGAU20(XL,XU,GRAND,ANSWER)
           PALL=PALL+ANSWER
C ELSE TEST FOR CONVERGENCE
           IF(DABS(ANSWER)*1.D6.LT.DABS(PALL))GOTO 20
           IF(PALL.EQ.0.D0.AND.INTERV.GE.KONSTA)GOTO 20
           XL=XU
           INTERV=INTERV+1
           GOTO 10
C THE TWO-TERM APPROXIMATION
20         ADD=0.D0
           XL=0.D0
           INTERV=1
30         XU=XL+20.D0
           INVOKE=0
           CALL DGAU20(XL,XU,SECOND,ANSWER)
           ADD=ADD+ANSWER
           IF(DABS(ANSWER)*1.D6.LT.DABS(ADD))GOTO 40
           IF(ADD.EQ.0.D0.AND.INTERV.GE.KONSTA)GOTO 40
35         XL=XU
           INTERV=INTERV+1
           GOTO 30
40         PALL2=ADD
           RETURN
           END
```

```
              DOUBLE PRECISION FUNCTION GRAND(X)
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              LOGICAL*1 TEST,TESTI,AUFL,QUFL,NOSAVE
              COMMON /IGP/ PLAM1,SHNID1,LPASS,LPM1,BSO16,NOSAVE
              COMMON /VDX/ INTERV,LSH,INVOKE
              COMMON /CONVRG/ QUFL,AUFL
              INVOKE=INVOKE+1
              TESTI=INTERV.LE.105
              TEST=TESTI
              SHNID=DOSHNI(INVOKE,INTERV,LSH,TEST,TESTI,LPASS,
          $     SHNID1,X,NOSAVE)
              ALAM=PLAM1
              BARG=DSQRT(X*ALAM)
              CALL DXBESI(BARG,LPM1,BANS,KODE)
              IF(KODE.NE.0)WRITE(5,1)KODE
    1         FORMAT(' SECOND/AVCHI(1). DXBESI KODE = ',I1)
              XOA=X/ALAM
              S=DSQRT(XOA)
              AVCHI2=0.5D0*DEXP(BARG-0.5D0*(X+ALAM))*DXI(S,LPM1)*BANS
              GRAND=AVCHI2*SHNID
              IF(INVOKE.EQ.20.D0) THEN
                AUFL=AVCHI2.EQ.0.D0
                QUFL=SHNID.EQ.0.D0
              ELSE
                AUFL=.FALSE.
                QUFL=.FALSE.
              END IF
              RETURN
              END
```



```
              DOUBLE PRECISION FUNCTION SECOND(X)
      C INTEGRAND FOR 2-TERM APPROXIMATION; MAKES USED OF SAVED
      C Q FUNCTIONS FROM THE 1-TERM FORMULA.
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              LOGICAL*1 TEST,TESTI,NOSAVE
              COMMON /IGP/ PLAM1,SHNID1,LPASS,LPM1,BSO16,NOSAVE
              COMMON /VDX/ INTERV,LSH,INVOKE
              INVOKE=INVOKE+1
              TESTI=INTERV.LE.105
              TEST=TESTI
              SHNID=DOSHNI(INVOKE,INTERV,LSH,TEST,TESTI,LPASS,
          $     SHNID1,X)
              ALAM=PLAM1
              BARG=DSQRT(X*ALAM)
              CALL DXBESI(BARG,LPM1,BANS,KODE)
              IF(KODE.NE.0)WRITE(5,1)KODE
    1         FORMAT(' SECOND/AVCHI(1). DXBESI KODE = ',I1)
              CALL DXBESI(BARG,LPASS,BANS2,KODE)
              IF(KODE.NE.0)WRITE(5,2)KODE
    2         FORMAT(' SECOND/AVCHI(2). DXBESI KODE = ',I1)
              XOA=X/ALAM
              S=DSQRT(XOA)
              AVCHI2=0.5D0*DEXP(BARG-0.5D0*(X+ALAM))*DXI(S,LPASS)*
          $ ( (1.D0+BSO16*(1.D0+XOA))*BANS/S
          $   -2.D0*BSO16*(1.D0+LPASS/ALAM)*BANS2 )
              SECOND=AVCHI2*SHNID
              RETURN
              END
```

```
0001            DOUBLE PRECISION FUNCTION DOSHNI(INVOKE,INTERV,LSH,TST,
             $ TESTI,LPASS,SHNID1,X)
        C SUBROUTINE TO CALCULATE & SAVE GENERALIZED Q-FUNCTIONS.
        C MUST BE DONE THIS WAY BECAUSE OF VIRTUAL ARRAY.
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            LOGICAL*1 SV(20,105,0:4),TEST,TESTI,TST
0004            VIRTUAL SH(20,105,0:4)
0005            COMMON /VALIDS/ SV
0006            SAVE SH
0007            TEST=TST
0008            IF(TEST)TEST=SV(INVOKE,INTERV,LSH)
0009            IF(TEST) THEN
0010              DOSHNI=SH(INVOKE,INTERV,LSH)
0011            ELSE
0012              DOSHNI=PNXY(LPASS,SHNID1,0.5D0*X,1.D-14)
0013              IF(TESTI) THEN
0014                SH(INVOKE,INTERV,LSH)=DOSHNI
0015                SV(INVOKE,INTERV,LSH)=.TRUE.
0016              END IF
0017            END IF
0018            RETURN
0019            END
```

APPENDIX 8G

COMPUTER PROGRAM TO COMPUTE BIT ERROR PROBABILITY FOR MFSK/FH WITH L HOPS/SYMBOL AND AT MOST ONE JAMMING TONE PER M-ARY BAND IN THE PRESENCE OF BOTH THERMAL NOISE AND TONE JAMMING USING EXACT FORMULATION

The following pages contain a listing of the FORTRAN-77 program used to obtain numerical values for the probability of bit error for MFSK/FH with L=1 hop/symbol and at most one jamming tone per M-ary band in the presence of thermal noise, using special-case exact formulas.

For a listing of the subprogram DBINCO, see Appendix 4F, listing page 8. For a listing of the subprogram DXBESI, see Appendix 4G, listing pages 12-13.

```
      PROGRAM M1HOP
C
C MFSK, 1 HOP/SYMBOL USING RESULTS IN PAPER BY MASSARO
C ASSUMING BARRAGE JAMMING WITH n = M
C
C PROGRAMMER: R. H. FRENCH, 21 FEBRUARY 1984
C
C VERSION 1.2.3
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /SIZE/ NO, NJ, NG, NK
      COMMON /INPUTS/ DEBNOL(3), DEBNJL(30), GAMLST(10),
     $ KLIST(3), NSLOTS
      CALL GET
      SLOTS=NSLOTS
      DO 900 IK=1,NK
      K=KLIST(IK)
      FK=K
      M=2**K
      FM=M
      W2B=FM*0.5D0/(FM-1.D0)
      DO 800 IG=1,NG
      GAMMA=GAMLST(IG)
      Q=GAMMA*SLOTS
      DENOM=SLOTS-FM+1.D0
      P00=(SLOTS-FM*Q-FM+1.D0)/DENOM
      P01=(FM-1.D0)*Q/DENOM
      P10=Q/DENOM
      DO 700 IO=1,NO
      EBNO=10.D0**(DEBNOL(IO)/10.D0)
      ESNO=FK*EBNO
      CALL PUT1(M,GAMMA,DEBNOL(IO))
      DO 600 IJ=1,NJ
      EBNJ=10.D0**(DEBNJL(IJ)/10.D0)
      ESNJ=FK*EBNJ
      RJN=ESNO/(GAMMA*ESNJ)
      CALL PROBS(ESNO,RJN,M,W2B,PE00,PE01,PE10)
      PE=P00*PE00+P01*PE01+P10*PE10
      CALL PUT2(DEBNJL(IJ),PE,PE00,PE01,PE10)
600   CONTINUE
700   CONTINUE
800   CONTINUE
900   CONTINUE
      STOP 0
      END
```

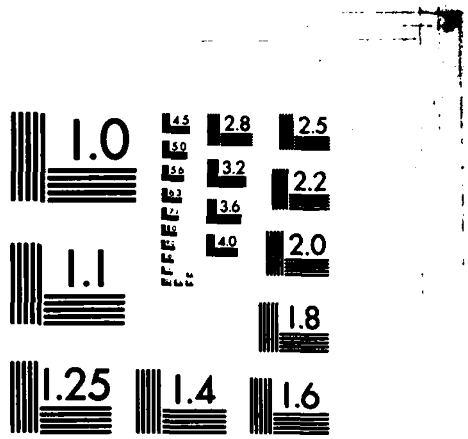


```
      SUBROUTINE GET
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION DGAM(10),DFO(5,3)
      COMMON /SIZE/ NO, NJ, NG, NK
      COMMON /INPUTS/ DEBNOL(3), DEBNJL(30), GAMLST(10),
     $ KLIST(3), NSLOTS
      DATA DGAM/ 4.1666666666666667D-4, 8.3333333333333333D-4,
     $ 0.005D0, 0.01D0, 0.1D0, 0.5D0, 0.00125D0, 0.0025D0,
     $ 0.0075D0, 0.05D0/
      DATA DFO /
     $   13.3525D0, 10.6065D0, 9.0939D0, 8.0783D0, 7.3295D0,
     $   12.3133D0,  9.6284D0, 8.1690D0, 7.1996D0, 6.4910D0,
     $   10.9444D0,  8.3524D0, 6.9718D0, 6.0696D0, 5.4183D0/
1     WRITE(5,2)
2     FORMAT(' HOW MANY VALUES OF K? [3] ',$)
      READ(5,3,ERR=1)NK
3     FORMAT(I1)
      IF(NK.EQ.0)NK=3
      IF(NK.LT.0.OR.NK.GT.3)GOTO 1
      DO 7 IN=1,NK
4     WRITE(5,5)IN,IN
5     FORMAT('  ENTER K(',I1,') [',I1,']: ',$)
      READ(5,6,ERR=4)KLIST(IN)
6     FORMAT(I1)
      IF(KLIST(IN).EQ.0)KLIST(IN)=IN
      IF(KLIST(IN).LT.0.OR.KLIST(IN).GT.5)GOTO 4
7     CONTINUE
8     WRITE(5,9)
9     FORMAT(' HOW MANY HOPPING SLOTS? [2400] ',$)
      READ(5,10,ERR=8)NSLOTS
10    FORMAT(I5)
      IF(NSLOTS.EQ.0)NSLOTS=2400
      IF(NSLOTS.LE.1)GOTO 8
11    WRITE(5,12)
12    FORMAT(' HOW MANY VALUES OF GAMMA? [5] ',$)
      READ(5,13,ERR=11)NG
13    FORMAT(I2)
      IF(NG.EQ.0)NG=5
      IF(NG.LT.0.OR.NG.GT.10)GOTO 11
      GAMMIN=1.D0/NSLOTS
      DO 17 IN=1,NG
14    WRITE(5,15)IN,DGAM(IN)
15    FORMAT('   ENTER GAMMA(',I2,') [',1PD10.3,']: ',$)
      READ(5,16,ERR=14)GAMLST(IN)
16    FORMAT(D10.3)
      IF(GAMLST(IN).EQ.0.D0)GAMLST(IN)=DGAM(IN)
      IF(GAMLST(IN).LT.GAMMIN.OR.GAMLST(IN).GT.1.D0)GOTO 14
17    CONTINUE
18    WRITE(5,19)
19    FORMAT(' HOW MANY VALUES OF EB/NJ? [11] ',$)
      READ(5,20,ERR=18)NJ
20    FORMAT(I2)
```

AD-A147 766 OPTIMUM JAMMING EFFECTS ON FREQUENCY-HOPPING M-ARY FSK 1/1
(FREQUENCY-SHIFT K..(U) LEE (J S) ASSOCIATES INC
ARLINGTON VA J S LEE ET AL. OCT 84 JC-2025-N

UNCLASSIFIED N00014-83-C-0312 F/G 17/4 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

PDP-11 FORTRAN-77 V4.0-1 13:52:01 8-May-84 Page 3

```
              IF(NJ.EQ.0)NJ=11
              IF(NJ.LT.0.OR.NJ.GT.30)GOTO 18
              DO 24 IN=1,NJ
              IF (NJ.LE.11) THEN
                D=5*(IN-1)
              ELSE
                IF (IN.EQ.1) THEN
                  D=1.D0
                ELSE
                  D=DEBNJL(IN-1)+1.D0
                END IF
              END IF
 21           WRITE(5,22)IN,D
 22           FORMAT('   ENTER EB/NJ(',I2,') [',F4.1,']: ',$)
              READ(5,23,ERR=21)DEBNJL(IN)
              IF(DEBNJL(IN).EQ.0.0D0)DEBNJL(IN)=D
 23           FORMAT(F5.1)
 24           CONTINUE
 25           WRITE(5,26)
 26           FORMAT(' HOW MANY VALUES OF EB/NO? [1] ',$)
              READ(5,20,ERR=18)NO
 27           FORMAT(I2)
              IF(NO.EQ.0)NO=1
              IF(NO.LT.0.OR.NO.GT.3)GOTO 18
              DO 31 IN=1,NO
 28           WRITE(5,29)IN,DFO(KLIST(1),IN)
 29           FORMAT('   ENTER EB/NO(',I2,') [',F7.4,']: ',$)
              READ(5,30,ERR=28)DEBNOL(IN)
 30           FORMAT(F5.1)
              IF(DEBNOL(IN).EQ.0.D0) DEBNOL(IN)=DFO(KLIST(1),IN)
 31           CONTINUE
              RETURN
              END
```

PDP-11 FORTRAN-77 V4.0-1 13:52:18 8-May-84 Page 4

```
              SUBROUTINE PUT1(M,GAMMA,DEBNO)
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              WRITE(6,1)M, M, GAMMA, DEBNO
 1            FORMAT('1',5X,I2,'-ARY FSK, 1 HOP/BIT, USING ',
            $ 'FORMULAS FROM MASSARO''S PAPER'/
            $ ' BARRAGE JAMMING WITH SPACING OF ',I2,
            $ ' SLOTS BETWEEN TONES'//' GAMMA = ',1PD10.3,5X,
            $ ' EB/NO = ',0PF7.4,' dB')
              WRITE(6,2)
 2            FORMAT(1X,'EB/NJ',5X,'P(E)',8X,'PE00',8X,'PE01',8X,
            $ 'PE10')
              RETURN
              END
```

PDP-11 FORTRAN-77 V4.0-1 13:52:20 8-May-84 Page 5

```
              SUBROUTINE PUT2(DEBNJ,PE,PE00,PE01,PE10)
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              WRITE(6,1)DEBNJ,PE,PE00,PE01,PE10
 1            FORMAT(1X,F5.1,2X,1P4D12.3)
              RETURN
              END
```

```
      SUBROUTINE PROBS(ESNO,RJN,M,W2B,PE00,PE01,PE10)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C COMPUTE PE00 USING RESULT FROM STEIN & JONES,
C   EQ. 14-45
C
      MM1=M-1
      PE00=0.D0
      POWM=-1.D0
      DO 10 K=1,MM1
      FK=K
      POWM=-POWM
      COEF=DBINCO(MM1,K)*POWM/(K+1)
      TERM=COEF*DEXP(-FK*ESNO/(FK+1.D0))
      PE00=PE00+TERM
10    CONTINUE
      PE00=W2B*PE00
C
C COMPUTE PE01 USING MASSARO'S EQUATION 16
C
      MM2=M-2
      POWK=1.D0
      PE01 = 1.D0
      DO 20 K=0,MM2
      FK=K
      FK1=1.D0+FK
      FK2=2.D0+FK
      A=RJN*FK1/FK2
      B=ESNO/(FK1*FK2)
      POWK=-POWK
      COEF=POWK*DBINCO(MM2,K)/FK1
      X=DEXP(-FK*FK2*B)
      BESARG=2.D0*DSQRT(A*B)
      CALL DXBESI(BESARG,0,BESSEL,KODE)
      IF(KODE.NE.0)WRITE(5,1)KODE
1     FORMAT(' DXBESI ERROR KODE = ',I1)
      QA=DSQRT(A+A)
      QB=DSQRT(B+B)
      PART=1.D0-Q(QA,QB)+DEXP(BESARG-A-B)*BESSEL/FK2
      TERM=COEF*X*PART
      PE01=PE01+TERM
20    CONTINUE
      PE01=W2B*PE01
C
C COMPUTE PE10 USING MASSARO'S EQ. 11
C
      PE10 = 0.D0
      BPART=2.D0*DSQRT(ESNO*RJN)
      XPART=ESNO+RJN
      POWK=-1.D0
      DO 30 K=1,MM1
      POWK=-POWK
```



```
      COEF=POWK*DBINCO(MM1,K)
      FK=K
      FK1=FK+1.D0
      COEF=COEF/FK1
      FKR=K/FK1
      BARG=BPART*FKR
      CALL DXBESI(BARG,0,BSL,KODE)
      IF(KODE.NE.0)WRITE(5,2)KODE
2     FORMAT(' DXBESI (SECOND CALL) ERROR KODE = ',I1)
      TERM=COEF*(BSL*DEXP(BARG-XPART*FKR))
      PE10=PE10+TERM
30    CONTINUE
      PE10=W2B*PE10
      RETURN
      END
```

APPENDIX 8H

COMPUTER PROGRAM TO COMPUTE BIT ERROR PROBABILITY FOR MFSK/FH WITH L HOPS/SYMBOL IN THE PRESENCE OF BOTH THERMAL NOISE AND BARRAGE TONE JAMMING USING EXACT FORMULATION

The following pages contain the listing of a computer program written in FORTRAN-77 to compute the bit error probability for MFSK/FH with L hops/bit in the presence of both thermal noise and barrage tone jamming, using the exact analytical formulation. To adapt the program to other tone jamming models, one need only replace the calculations in the subroutine PR1HOP with the appropriate one-hop jamming event probabilities for whatever jamming model is desired.

For a listing of the function DXI which is used in this program, see Appendix 4I. The subroutine DCEL1 computes the complete elliptic integral of the first kind. It is a double-precision adaptation of CEL1 from the Digital Equipment Corporation Scientific Subroutine Package [19]. For function PNXY, see Appendix 8I.

This program makes considerable reuse of temporary storage areas and a large virtual array to avoid a complicated overlay structure.

```
0001          PROGRAM TJMLH
        C
        C THIS PROGRAM COMPUTES THE ERROR PROBABILITY FOR M-ARY FSK/FH
        C WITH MULTIPLE HOPS/BIT IN THE PRESENCE OF BARRAGE JAMMING
        C
        C ANALYSIS: L. E. MILLER, R. H. FRENCH
        C PROGRAM: R. H. FRENCH
        C
        C 26 OCT 1983, 3 NOV 83, 8 NOV 83, 25 JAN 84, 26 JAN 84,
        C 7 FEB 84, 11 APR 84, 17 APR 84, 1 MAY 84, 2 MAY 84, 3 MAY 84
        C
        C V 5.3.1
        C
        C SPEED-UP VIA SAVED SIGNAL-CHANNEL DENSITY FUNCTION VALUES
        C IN THE INTEGRAND FUNCTIONS AND SAVED PHASE-DIFFERENCE DENSITIES
        C
0002          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003          LOGICAL*1 GOOD
        C COMMON /INPUTS/ PASSES PARAMETERS OF THE RUN
0004          COMMON /INPUTS/ DEBNOL(5),DEBNJL(11),LLIST(4),NSLOTS,
           $    NSTEP,GAMLST(31),K,MM
        C COMMON /SIZE/ PASSES NUMBERS OF PARAMETERS
0005          COMMON /SIZE/ NO,NJ,NL,NG
0006          CALL GET
        C INITIALIZE THE PHASE-DIFFERENCE DENSITY UP TO MIN(MAX(L),6)
        C SINCE WE ONLY HAVE STORAGE FOR UP TO 6 VALUES
0007          LMAX=LLIST(1)
0008          IF(NL.GE.2) THEN
0009            DO 100 IL=2,NL
0010            IF(LMAX.LT.LLIST(IL))LMAX=LLIST(IL)
0011    100     CONTINUE
0012          END IF
0013          IF(LMAX.GT.6)LMAX=6
0014          DO 200 IL=1,LMAX
0015          CALL SETPZE(IL,IL)
0016    200   CONTINUE
0017          DO 900 IL=1,NL
0018          LL=LLIST(IL)
0019          LSAVE=LL
0020          FLL=LL
0021          DO 800 IO=1,NO
0022          EBNO=10.DO**(DEBNOL(IO)/10.DO)
0023          SNR=K*EBNO/FLL
0024          DO 700 IG=1,NG
0025          GAMMA=GAMLST(IG)
        C ADD 0.5 TO GET NEAREST INTEGER
0026          NTONES=GAMMA*NSLOTS+0.5DO
0027          CALL PUT1(MM,LL,NSLOTS,NSTEP,DEBNOL(IO),GAMMA)
0028          CALL GENPIE(LL,MM,NTONES,NSTEP,NSLOTS,GOOD)
0029          IF(.NOT.GOOD) GOTO 700
```



```
0030          DO 600 IJ=1,NJ
0031          R=10.DO**(DEBNJL(IJ)/10.DO)
0032          RJN=EBNO/(GAMMA*R)
        C EVALUATE THE PROBABILITY
0033          CALL PSUBE(SNR,RJN,LL,MM,PE)
        C WRITE IT TO PRINT FILE
0034          CALL PUT2(DEBNJL(IJ),PE)
0035    600   CONTINUE
0036    700   CONTINUE
0037    800   CONTINUE
0038    900   CONTINUE
0039          STOP 0
0040          END
```

```
      SUBROUTINE GET
C
C  INTERACTIVE INPUT OF PARAMETERS FOR RUN
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      CHARACTER*7 FIELD,BLANK7
C COMMON /ZETDEN/ IS SHARED WITH STORAGE OF ZETA DENSITY.  IT IS
C USED TO HOLD (TEMPORARILY) THE DEFAULT LISTS FOR THE INTERACTIVE
C PARAMETER INPUT, AFTER WHICH THE DEFAULT LISTS CAN BE DISCARDED
C AND THE STORAGE USED FOR THE ZETA DENSITY.  THE DUMMY ARRAY AT
C THE END OF THE COMMON BLOCK IS REQUIRED TO MAKE FIRST OCCURRENCE
C AT LEAST AS BIG AS SUBSEQUENT OCCURRENCES.
      COMMON /ZETDEN/ DG(31),DSNR(5,4),DUMMY(89)
      COMMON /INPUTS/ DEBNOL(5),DEBNJL(11),LLIST(4),NSLOTS,
     $  NSTEP,GAMLST(31),K,MM
      COMMON /SIZE/ NO,NJ,NL,NG
      DATA BLANK7/'       '/
32    WRITE(5,33)
33    FORMAT(' ENTER BITS/SYMBOL (K) [2]: ',$)
      READ(5,3,ERR=32)K
      IF(K.EQ.0)K=2
      IF(K.LE.0)GOTO 32
      MM=2**K
1     WRITE(5,2)
2     FORMAT(' HOW MANY VALUES OF EB/NO? [1]: ',$)
      READ(5,3,ERR=1)NO
3     FORMAT(I2)
      IF(NO.EQ.0)NO=1
      IF(NO.LE.0.OR.NO.GT.5) GOTO 1
      DO 7 IN=1,NO
      IF(K.LE.4) THEN
        DO=DSNR(IN,K)
      ELSE
        DO=0.DO
      END IF
4     WRITE(5,5)IN,DO
5     FORMAT('   ENTER EB/NO(',I2,') [',F7.4,']: ',$)
      READ(5,6,ERR=4)FIELD
6     FORMAT(A7)
      IF(FIELD.EQ.BLANK7) THEN
        DEBNOL(IN)=DO
      ELSE
        DECODE(7,61,FIELD,ERR=4)DEBNOL(IN)
61      FORMAT(F7.4)
      END IF
7     CONTINUE
8     WRITE(5,9)
9     FORMAT(' HOW MANY VALUES OF EB/NJ? [11]: ',$)
      READ(5,3,ERR=8)NJ
      IF(NJ.EQ.0)NJ=11
      IF(NJ.LE.0.OR.NJ.GT.11)GOTO 8
```



```
      DO 14 IN=1,NJ
      DJ=5*(IN-1)
11    WRITE(5,12)IN,DJ
12    FORMAT('   ENTER EB/NJ(',I2,') [',F4.1,']: ',$)
      READ(5,13,ERR=11)DEBNJL(IN)
13    FORMAT(F4.1)
      IF(DEBNJL(IN).EQ.0.DO)DEBNJL(IN)=DJ
14    CONTINUE
15    WRITE(5,16)
16    FORMAT(' HOW MANY VALUES OF L? [4]: ',$)
      READ(5,3,ERR=14)NL
      IF(NL.EQ.0)NL=4
      IF(NL.LE.0.OR.NL.GT.4)GOTO 15
      DO 21 IN=1,NL
18    WRITE(5,19)IN,IN
19    FORMAT('   ENTER L(',I1,') [',I1,']: ',$)
      READ(5,3,ERR=18)LLIST(IN)
      IF(LLIST(IN).EQ.0)LLIST(IN)=IN
      IF(LLIST(IN).LE.0)GOTO 18
21    CONTINUE
22    WRITE(5,23)
23    FORMAT(' HOW MANY HOPPING SLOTS? [2400]: ',$)
      READ(5,24,ERR=22)NSLOTS
24    FORMAT(I5)
      IF(NSLOTS.EQ.0)NSLOTS=2400
      IF(NSLOTS.LE.1)GOTO 22
50    WRITE(5,51)
51    FORMAT(' ENTER SPACING BETWEEN TONES [1]: ',$)
      READ(5,3,ERR=50)NSTEP
      IF(NSTEP.EQ.0)NSTEP=1
      IF(NSTEP.LE.0.OR.NSTEP.GT.MM) GOTO 50
25    WRITE(5,26)
26    FORMAT(' HOW MANY VALUES OF GAMMA? [5]: ',$)
      READ(5,3,ERR=25)NG
      IF(NG.EQ.0)NG=5
      IF(NG.LE.0.OR.NG.GT.31)GOTO 25
      DO 31 IN=1,NG
28    WRITE(5,29)IN,DG(IN)
29    FORMAT('   ENTER GAMMA(',I2,') [',1PD8.1,']: ',$)
      READ(5,30,ERR=28)GAMLST(IN)
30    FORMAT(D15.8)
      IF(GAMLST(IN).EQ.0.DO)GAMLST(IN)=DG(IN)
      IF(GAMLST(IN).LE.0.DO.OR.GAMLST(IN).GT.1.DO)GOTO 28
31    CONTINUE
      RETURN
      END
```

```
0001            SUBROUTINE PUT1(M,L,NSLOTS,NSTEP,DEBNO,GAMMA)
        C
        C WRITE PAGE HEADERS
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            WRITE(6,1)M,L,NSLOTS,NSTEP,DEBNO,GAMMA
0004    1       FORMAT('1',I2,'-ARY FSK/FH (EXACT ANALYSIS)'/
            $ ' L = ',I1,' HOPS/BIT ',I5,
            $ ' HOPPING SLOTS'/' JAMMING TONES SPACED ',I5,' SLOTS'/
            $ ' EB/NO = ',F8.4,' DB'/' GAMMA = ',1PD10.3//
            $ '  EB/NJ',5X,'P(E)')
0005            RETURN
0006            END
```



```
0001            SUBROUTINE PUT2(DEBNJ,PE)
        C
        C WRITE A LINE OF RESULTS
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            WRITE(6,1)DEBNJ,PE
0004    1       FORMAT(1X,F6.2,2X,1PD10.3)
0005            RETURN
0006            END
```



```
0001            SUBROUTINE PSUBE(SNR,RJN,LL,M,PE)
        C
        C COMPUTE UNCONDITIONAL ERROR PROBABILITY
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            INTEGER*4 JAM(8),LOW,M4,LUP(8),JSUB(8),ISUB,I200,IPSUB,NPS,
            $     LINC,JAM1
0004            LOGICAL*1 SV,GO,NONE,STORE
        C WE DO WANT TO STORE ZERO ELEMENTS OF THE DENSITY FUNCTION,
        C SINCE IT SAVES TIME TO AVOID REPEATING THE UNDERFLOWS
0005            DATA STORE/.TRUE./
        C COMMONS /SHARE/ AND /SHARE2/ SAVE ADDRESS SPACE
        C BY RE-USING TEMPORARY STORAGE NEEDED ONLY LOCALLY.
        C  /SHARE/ SHARES PRERR & IPSUB WITH C AND ICSUB IN GENPIE
        C  /SHARE2/ IS COMMONLY NEEDED 4-BYTE INTEGER CONSTANT ARRAYS,
        C           INITIALIZED VIA A BLOCK DATA SUBPROGRAM
0006            COMMON /SHARE/ PRERR(200),IPSUB(200)
0007            COMMON /SHARE2/ LOW(8),LINC(8),I200
        C
        C **** WARNING ****
        C       COMMON BLOCK /VALIDS/ IS SHARED BY THE SV(20,105) ARRAY
        C       AND ARRAYS A(100) AND IASUB(100) TO SAVE ADDRESS SPACE
        C
        C       GENPIE IS CALLED ONLY FROM THE MAIN PROGRAM, AND NEEDS
        C       A(100) AND IASUB(100) ONLY AS WORK ARRAYS TO GENERATE
        C       FINAL RESULTS IN ARRAY D(200) IN COMMON BLOCK /EVENTS/.
        C
        C       PSUBE IS CALLED AFTER GENPIE IS DONE, SO IT CAN SAFELY
        C       RE-USE THE STORAGE AS THE ARRAY OF VALIDITY INDICATORS
        C       FOR THE SAVED PDF SAMPLES FOR NUMERICAL INTEGRATION.
        C
        C       PSUBE WILL BE DONE FOR ANY ONE CASE BEFORE
        C       GENPIE NEEDS THE WORK SPACE AGAIN FOR THE NEXT CASE.
        C
        C       THIS IS ADMITTEDLY POOR PROGRAMMING PRACTICE, BUT THE
        C       ADDRESS SPACE IS NEEDED TO FREE UP AN APR TO MAP TO THE
        C       VIRTUAL ARRAY SPDF WHICH IS USED TO SAVE UP THE PROBA-
        C       BILITIES FOR POSSIBLE RE-USE TO SAVE A LOT OF TIME.
        C
0008            COMMON /VALIDS/ SV(20,105)
0009            PE=0.D0
0010            M4=M
0011            NPS=0
0012            DO 10 I=1,M
0013            LUP(I)=LL
0014    10      CONTINUE
0015            JAM1=-1
        C START VECTOR-INDEXED LOOP
0016            CALL VLINIT(JAM,LOW,M4)
0017    100     CONTINUE
```

```
0018            IF(JAM1.NE.JAM(1)) THEN
        C UPDATE TEST VALUE FOR NEXT TIME, AND ...
0019              JAM1=JAM(1)
        C ... RECOMPUTE THE DENSITY OF SUM OF COSINES, IF NEEDED
0020              IF(JAM1.GT.6) CALL SETPZE(JAM1,7)
        C ... AND MARK SAVED SIGNAL-CHANNEL PDF AS INVALID
0021              DO 4 LB=1,105
0022              DO 4 LA=1,20
0023    4         SV(LA,LB)=.FALSE.
0024            END IF
0025            CALL EVENT(LL,M4,JAM,PIE)
        C BYPASS CONDITIONAL ERROR PROBABILITY CALCULATION IF EVENT
        C PROBABILITY IS ZERO.  THIS SAVES MUCH TIME.
0026            IF(PIE.EQ.0.D0)GOTO 101
        C SINCE JAMMING PROBABILITIES DEPEND ONLY ON NO. OF CHANNELS
        C HAVING JAM(I) HOPS JAMMED AND NOT THE ARRANGEMENT OF THE
        C CHANNELS, WE CAN SORT THE NON-SIGNAL CHANNELS INTO ASCENDING
        C NUMBERS OF HOPS JAMMED.  THIS REDUCES NUMBER OF DISTINCT
        C CONDITIONAL ERROR PROBABILITIES WHICH MUST BE SAVED TO AVOID
        C RECOMPUTING THEM UNNECESSARILY.
0027            CALL SORTJ(JAM,JSUB,M)
        C STORE AS ELEMENT OF A SPARSE ARRAY TO SAVE MEMORY
        C EVEN THOUGH WE STORE ZEROS, THE SORTING OF SUBSCRIPTS
        C CUTS OUT MANY ELEMENTS.
0028            CALL LOCN(M4,LOW,LUP,JSUB,ISUB)
        C TRY TO FIND CONDITIONAL ERROR PROBABILITY IN STORED ARRAY
0029            CALL LOOKUP(PALL,PRERR,IPSUB,NPS,I200,ISUB,STORE,NONE)
        C IF IT IS NOT THERE, WE MUST COMPUTE IT
0030            IF(NONE) THEN
0031              CALL PELL(SNR,RJN,LL,JSUB,M,PALL)
        C  ... AND SAVE IT FOR POSSIBLE FUTURE RE-USE
0032              CALL PUTIN(PALL,PRERR,IPSUB,NPS,I200,ISUB,KODE,STORE)
0033              IF(KODE.NE.0) STOP 'NO ROOM FOR PR(E|JAM)'
0034            END IF
        C SUM UP UNCONDITIONAL ERROR PROBABILITY
0035            UEP=PIE*PALL
0036            PE=PE+UEP
        C ITERATE THE VECTOR-INDEX LOOP
0037    101     CALL VLITER(JAM,LOW,LUP,LINC,M4,GO)
0038            IF(GO) GOTO 100
0039            RETURN
0040            END
```



```
0001            SUBROUTINE EVENT(LL,M,JAM,PIE)
        C
        C SUBROUTINE TO LOOK UP EVENT PROBABILITY FROM STORED ARRAY
        C
0002            IMPLICIT INTEGER*4(I-N), DOUBLE PRECISION(A-H,O-Z)
0003            INTEGER*2 LL,I,M2
0004            LOGICAL*1 STORE,NONE
0005            DIMENSION JAM(8),LUP(8)
        C COMMON /EVENTS/ PASSES PROBABILITIES COMPUTED BY GENPIE
0006            COMMON /EVENTS/ D(200),IDSUB(200),NUSED
0007            COMMON /SHARE2/ LOW(8),LINC(8),I200
0008            DATA STORE/.FALSE./
0009            LLL=LL
0010            M2=M
        C SET UP ARRAY DESCRIPTION D(0:LL,...,0:LL) WITH M DIMENSIONS
0011            DO 1 I=1,M2
0012            LUP(I)=LLL
0013    1       CONTINUE
        C COMPUTE LINEAR EQUIVALENT SUBSCRIPT FOR JAMMING EVENT
0014            CALL LOCN(M,LOW,LUP,JAM,ISUB)
        C LOOK UP THE VALUE, GET 0.D0 IF NOT THERE
0015            CALL LOOKUP(PIE,D,IDSUB,NUSED,I200,ISUB,STORE,NONE)
0016            RETURN
0017            END
```

```
0001          SUBROUTINE PELL(SNR,RJN,LL,JAM,M,PALL)
        C
        C COMPUTE CONDITIONAL ERROR PROBABILITY, GIVEN A JAMMING EVENT
        C
0002          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003          INTEGER*4 JAM(8)
0004          LOGICAL*1 NOSAVE,QUFL,AUFL,LEFT,RIGHT
0005          EXTERNAL GRAND
        C COMMON /IGP/ PASSES PARAMETERS TO INTEGRAND FUNCTION
0006          COMMON /IGP/ PLAM1,SJNR,LJAM,LPASS,LPM1,NOSAVE
        C COMMON /JAMCON/ PASSES JAMMING PARAMETERS TO INTEGRAND
0007          COMMON /JAMCON/IPQ(7),FLJ(7),NJP
        C COMMON /VDX/ PASSES INFORMATION NEEDED TO DETERMINE IF SAVED
        C   DENSITY IS AVAILABLE
0008          COMMON /VDX/ INTERV,INVOKE
        C COMMON /CONVRG/ RETURNS UNDERFLOW FLAGS FOR INTEGRATION
        C    CONVERGENCE LOGIC
0009          COMMON /CONVRG/ QUFL,AUFL
0010          FM=M
0011          W2B=0.5D0*FM/(FM-1.D0)
0012          LPASS=LL
0013          LPM1=LPASS-1
0014          FLL=LL
0015          FL1=JAM(1)
0016          LJAM=JAM(1)
        C COUNT UP POWERS OF Q-FUNCTIONS OF EACH DISTINCT ARGUMENT
0017          IPQ(1)=1
0018          FLJ(1)=JAM(2)
0019          NJP=1
0020          DO 9 KOUNT=2,7
0021          IPQ(KOUNT)=0
0022    9     CONTINUE
        C IF M .LE. 2, THEN THERE IS ONLY 1 Q-FUNCTION TERM
0023          IF(M.GT.2)THEN
0024            DO 12 KOUNT=3,M
0025            IF(JAM(KOUNT).NE.JAM(KOUNT-1)) THEN
0026              NJP=NJP+1
0027              FLJ(NJP)=JAM(KOUNT)
0028            END IF
0029            IPQ(NJP)=IPQ(NJP)+1
0030    12      CONTINUE
0031          END IF
        C CONVERT TO LAMBDA/2*LL VALUES
0032          DO 11 KOUNT=1,NJP
0033          FLJ(KOUNT)=FLJ(KOUNT)*RJN/FLL
0034    11    CONTINUE
        C SET CONSTANTS FOR INTEGRAND FUNCTION
0035          PLAM1=2.D0*(FLL*SNR+FL1*RJN)
0036          SJNR=2.D0*DSQRT(SNR*RJN)
0037          CONSTA=PLAM1
0038          NOSAVE=.FALSE.
```



```
        C ADAPT THE INTEGRATION STEP TO THE WIDTH OF THE INTEGRAND
        C AS MEASURED BY A PORTION OF THE NON-CENTRAL PARAMETER OF
        C THE SIGNAL CHANNEL DENSITY
0039          XINC=10.D0
0040          IF(CONSTA.GE.1000.D0) THEN
0041            NOSAVE=.TRUE.
0042            XINC=100.D0
0043            IF(CONSTA.GT.10000.D0) XINC=1000.D0
0044            IF(CONSTA.GT.1.D5) XINC=1.D4
0045            IF(CONSTA.GT.1.D6) XINC=1.D5
0046          END IF
0047    302   KONSTA=CONSTA/XINC+0.5D0
0048          SHNID1=FL2*RJN/FLL
0049          PALL=0.D0
0050          XL=0.D0
0051          INTERV=1
0052          IF(KONSTA.GT.100) THEN
        C WORK FROM THE MIDDLE OUT FOR LARGE NONCENTRAL PARAMETERS;
        C SO WE CAN'T USE THE SAVED SIGNAL-CHANNEL DENSITY ARRAY...
0053            NOSAVE=.TRUE.
0054            RIGHT=.FALSE.
0055            LEFT=.FALSE.
0056            INTERV=KONSTA
0057            LENGTH=INTERV
0058            XM=XINC*KONSTA
0059    111     XU=XM
0060            XL=XU-XINC
0061            INVOKE=0
0062            CALL DGAU20(XL,XU,GRAND,ANSWER)
        C IF ANSWER IS ZERO, WE ARE NOT IN THE REGION WE WANT TO BE IN
0063              IF(ANSWER.EQ.0.D0) THEN
        C IF 0 DUE TO Q FUNCTION UNDERFLOW, TRY FURTHER LEFT.  BUT IF
        C LAST MOVE WAS RIGHT, BACK UP HALF WAY AND TRY AGAIN
0064                IF(QUFL.AND.(.NOT.AUFL)) THEN
        C IF PREVIOUS STEP WAS RIGHT, BACK UP ONLY HALF WAY
0065                  IF(.NOT.LEFT)THEN
0066                    LENGTH=LENGTH/2
0067                    IF(LENGTH.EQ.0) GOTO 800
0068                  END IF
0069                  LEFT=.TRUE.
0070                  RIGHT=.FALSE.
0071                  INTERV=INTERV-LENGTH
        C IF WE CAN'T GO FURTHER LEFT, RETURN ZERO ANSWER
0072                  IF(INTERV.LE.0) GOTO 800
0073                  XM=INTERV*XINC
0074                  GOTO 111
        C IF PDF UNDERFLOWS, TRY FURTHER RIGHT; BUT IF LAST
        C MOVE WAS LEFT, BACK UP HALF WAY AND TRY AGAIN
0075                ELSE IF((.NOT.QUFL).AND.AUFL) THEN
0076                  IF(.NOT.RIGHT) THEN
0077                    LENGTH=LENGTH/2
0078                    IF(LENGTH.EQ.0) GOTO 800
0079                  END IF
```

```
0080                RIGHT=.TRUE.
0081                LEFT=.FALSE.
0082                INTERV=INTERV+LENGTH
0083                XM=XINC*INTERV
0084                GOTO 111
     C BOTH UNDERFLOW, RETURN 0 ANSWER
0085              ELSE IF(AUFL.AND.QUFL) THEN
0086                GOTO 800
0087              ELSE
     C ONLY PRODUCT UNDERFLOWED, SO STARTING POINT IS OK
0088                GOTO 700
0089              END IF
     C RETURN A ZERO ANSWER
0090  800         PALL=0.D0
0091              RETURN
0092              END IF
     C NOW WE HAVE A REGION WHERE SOMETHING NONZERO MAY ARISE, SO
     C DO THE INTEGRAL BOTH WAYS FROM HERE. LEFTWARD FIRST.
0093  700         PALL=ANSWER
0094              JNTERV=INTERV
0095              XU=XL
0096  701         XL=XU-XINC
0097              INTERV=INTERV-1
     C STOP AT THE ORIGIN.  SINCE XL IS AN INTEGER MULTIPLE OF XINC,
     C WE DON'T HAVE TO WORRY ABOUT A PARTIAL INTERVAL.
0098              IF(XL.LT.0.D0) GOTO 600
0099              INVOKE=0
0100              CALL DGAU20(XL,XU,GRAND,ANSWER)
0101              PALL=PALL+ANSWER
     C IF WE REACH ZERO TAIL, GO ON TO THE RIGHTWARD PART TO AVOID
     C WASTING TIME COMPUTING UNDERFLOWS
0102              IF(ANSWER.EQ.0.D0) GOTO 600
0103              GOTO 701
     C NOW SET UP TO LET THE NORMAL CODE TAKE OVER FOR RIGHTWARD PART
0104  600         XL=XM
0105              INTERV=JNTERV+1
0106            END IF
0107  10        XU=XL+XINC
0108            INVOKE=0
0109            CALL DGAU20(XL,XU,GRAND,ANSWER)
0110            PALL=PALL+ANSWER
     C TEST FOR CONVERGENCE...EITHER STRICT INEQUALITY
0111            IF(DABS(ANSWER)*1.D6.LT.DABS(PALL))GOTO 20
     C ...OR THE ENTIRE RESULT UNDERFLOWS TO ZERO
0112            IF(PALL.EQ.0.D0.AND.INTERV.GE.KONSTA)GOTO 20
0113            XL=XU
0114            INTERV=INTERV+1
0115            GOTO 10
     C CONVERT ANSWER FROM SYMBOL TO BIT ERROR PROBABILITY AND RETURN
0116  20        PALL=W2B*PALL
0117            RETURN
0118            END
```



```
0001          DOUBLE PRECISION FUNCTION GRAND(X)
     C
     C INTEGRAND FUNCTION FOR COMPUTATION OF CONDITIONAL ERROR PROBABILITY
     C
0002          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003          LOGICAL*1 SV,TST,AUFL,QUFL,NOSAVE
0004          LOGICAL*1 TEST,TESTI
0005          VIRTUAL SPDF(20,105)
     C COMMON /JAMCON/ IS INITIALIZED BY SUBROUTINE PELL
0006          COMMON /JAMCON/ IPQ(7),FLJ(7),NJP
     C
     C **** WARNING ****
     C        COMMON BLOCK /VALIDS/ IS SHARED BY THE SV(20,105) ARRAY
     C        AND ARRAYS A(100) AND IASUB(100) TO SAVE ADDRESS SPACE
     C
     C        GENPIE IS CALLED ONLY FROM THE MAIN PROGRAM, AND NEEDS
     C        A(100) AND IASUB(100) ONLY AS WORK ARRAYS TO GENERATE
     C        FINAL RESULTS IN ARRAY D(200) IN COMMON BLOCK /EVENTS/.
     C
     C        PSUBE IS CALLED AFTER GENPIE IS DONE, SO IT CAN SAFELY
     C        RE-USE THE STORAGE AS THE ARRAY OF VALIDITY INDICATORS
     C        FOR THE SAVED PDF SAMPLES FOR NUMERICAL INTEGRATION.
     C
     C        PSUBE WILL BE DONE FOR ANY ONE CASE BEFORE
     C        GENPIE NEEDS THE WORK SPACE AGAIN FOR THE NEXT CASE.
     C
     C        THIS IS ADMITTEDLY POOR PROGRAMMING PRACTICE, BUT THE
     C        ADDRESS SPACE IS NEEDED TO FREE UP AN APR TO MAP TO THE
     C        VIRTUAL ARRAY SPDF WHICH IS USED TO SAVE UP THE PROBA-
     C        BILITIES FOR POSSIBLE RE-USE TO SAVE A LOT OF TIME.
     C
0007          COMMON /VALIDS/ SV(20,105)
     C COMMON /CONVRG/ RETURNS CONVERGE TEST AIDS TO SUBROUTINE PELL
0008          COMMON /CONVRG/ QUFL,AUFL
     C COMMONS /IGP/ AND /VDX/ ARE INITIALIZED BY SUBROUTINE PELL;
     C          /VDX/ IS UPDATED IN THIS ROUTINE.
0009          COMMON /IGP/ PLAM1,SJNR,LJAM,LPASS,LPM1,NOSAVE
0010          COMMON /VDX/ INTERV,INVOKE
     C COMMON /ZETDEN/ IS DENSITY OF SIGNAL-CHANNEL SIGNAL-JAMMING
     C PHASE DIFFERENCE.  IT IS INITIALIZED THROUGH CALL TO SETPZE
0011          COMMON /ZETDEN/ PZETA(10,7),ZETA(10,7)
0012          INVOKE=INVOKE+1
     C CAN WE SAVE OR USE SAVED SIGNAL-CHANNEL DENSITY?
     C  1) IS IT WITHIN BOUNDS OF ARRAY?
0013          TESTI=INTERV.LE.105
0014          TEST=TESTI
     C  2) IS THE INTEGRAL BEING DONE IN NORMAL ORDER?
0015          TST=TEST.AND..NOT.NOSAVE
     C  IF BOTH CONDITIONS MET, LOOK UP THE VALIDITY FLAG
0016          IF(TST)TST=SV(INVOKE,INTERV)
     C  ELSE JUST KEEP IT FALSE
```

```
0017             IF(TST) THEN
        C  LOOK UP STORED DENSITY FUNCTION
0018               AVCHI2=SPDF(INVOKE,INTERV)
0019             ELSE
        C  MUST COMPUTE THE DENSITY FUNCTION
0020               IF(LJAM.NE.0) THEN
        C     AVERAGE THE SIGNAL-CHANNEL DENSITY OVER THE SUM OF
        C     COSINES OF PHASE DIFFERENCES. P(ZETA) STORED FOR LJAM<=6.
0021                 NDX=MIN0(LJAM,7)
0022                 AVCHI2=0.D0
0023                 DO 100 I=1,10
0024                 ALAMP=PLAM1+SJNR*ZETA(I,NDX)
0025                 ALAMM=PLAM1-SJNR*ZETA(I,NDX)
0026                 BARGP=DSQRT(X*ALAMP)
0027                 BARGM=DSQRT(X*ALAMM)
0028                 CALL DXBESI(BARGP,LPM1,BANSP,KODE)
0029                 IF(KODE.NE.0)WRITE(5,1)KODE
0030    1            FORMAT(' AVCHI-P DXBESI KODE = ',I1)
0031                 CALL DXBESI(BARGM,LPM1,BANSM,KODE)
0032                 IF(KODE.NE.0)WRITE(5,2)KODE
0033    2            FORMAT(' AVCHI-M DXBESI KODE = ',I1)
0034                 SP=DSQRT(X/ALAMP)
0035                 SM=DSQRT(X/ALAMM)
0036                 AVCH2P=0.5D0*DEXP(BARGP-0.5D0*(X+ALAMP))*DXI(SP,LPM1)*BANSP
0037                 AVCH2M=0.5D0*DEXP(BARGM-0.5D0*(X+ALAMM))*DXI(SM,LPM1)*BANSM
0038                 AVCHI2=AVCHI2+PZETA(I,NDX)*(AVCH2P+AVCH2M)
0039    100          CONTINUE
0040               ELSE
        C THERE ARE NO JAMMED HOPS, SO NO AVERAGING
0041                 ALAMP=PLAM1
0042                 BARGP=DSQRT(X*ALAMP)
0043                 CALL DXBESI(BARGP,LPM1,BANSP,KODE)
0044                 IF(KODE.NE.0)WRITE(5,3)KODE
0045    3            FORMAT(' 0 DXBESI KODE = ',I1)
0046                 SP=DSQRT(X/ALAMP)
0047                 AVCHI2=0.5D0*DEXP(BARGP-0.5D0*(X+ALAMP))*DXI(SP,LPM1)*BANSP
0048               END IF
        C SHOULD WE SAVE THE VALUE JUST COMPUTED?
0049               IF(TESTI.AND..NOT.NOSAVE) THEN
0050                 SPDF(INVOKE,INTERV)=AVCHI2
0051                 SV(INVOKE,INTERV)=.TRUE.
0052               END IF
0053             END IF
        C Q-FUNCTION PRODUCT FOR NON-SIGNAL CHANNELS
0054             QARG=0.5D0*X
0055             QP=1.D0
0056             DO 200 I=1,NJP
0057             QT=PNXY(LPASS,FLJ(I),QARG,1.D-13)
0058             QP=QP*DXI(1.D0-QT,IPQ(I))
0059    200      CONTINUE
0060             SHNID=1.D0-QP
```



```
        C COMPLETE DENSITY FUNCTION
0061             GRAND=AVCHI2*SHNID
        C IF THIS IS THE LAST POINT FROM THE GAUSSIAN QUADRATURE,
        C RETURN UNDERFLOW INDICATORS
0062             IF(INVOKE.EQ.20) THEN
0063               AUFL=AVCHI2.EQ.0.D0
0064               QUFL=SHNID.EQ.0.D0
0065             ELSE
0066               AUFL=.FALSE.
0067               QUFL=.FALSE.
0068             END IF
0069             RETURN
0070             END
```

```
0001            SUBROUTINE SETPZE(L1,NDX)
        C
        C SUBROUTINE TO INITIALIZE THE DENSITY FUNCTION FOR THE SUM OF THE
        C COSINES OF THE PHASE DIFFERENCES BETWEEN THE SIGNAL AND THE NOISE
        C
        C THIS SUBROUTINE USES THE FUNCTION PDFZET TO PERFORM THE ACTUAL
        C COMPUTATION OF THE DENSITY FUNCTION
        C
        C THE COMPUTED DENSITY FUNCTION VALUES ARE PRE-WEIGHTED BY THE
        C GAUSSIAN QUADRATURE INTEGRATION WEIGHTS TO SAVE TIME IN THE
        C SUBROUTINE GRAND WHICH USES GAUSSIAN QUADRATURE TO INTEGRATE
        C THE PRODUCT OF THIS DENSITY WITH THE (CONDITIONAL) NON-CENTRAL
        C CHI-SQUARED DENSITY OF THE SIGNAL CHANNEL SAMPLES.
        C THE SCALING TO THE INTERVAL (-LL,+LL) IS ALSO HANDLED IN THIS
        C SUBROUTINE.  ARRAY ZETA(10) IS THE SCALED ABSCISSAS FOR THE
        C GAUSSIAN QUADRATURE DONE IN SUBROUTINE GRAND.
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            INTEGER*4 L1
        C COMMON /ZETDEN/ IS RE-INITIALIZED BY THIS SUBROUTINE.  IT WAS
        C PREVIOUSLY USED AS TEMPORARY STORAGE FOR DEFAULT LISTS FOR "GET".
0004            COMMON /ZETDEN/ PZETA(10,7),ZETA(10,7)
        C COMMON /GQWTS/ SHARES GAUSSIAN QUADRATURE WEIGHTS
0005            COMMON /GQWTS/ X(10),W(10)
0006            SCALE=L1
0007            ANORM=0.DO
0008            DO 100 I=1,10
0009            ZETA(I,NDX)=SCALE*X(I)
0010            PZETA(I,NDX)=PDFZET(ZETA(I,NDX),L1)*W(I)*SCALE
0011            ANORM=ANORM+PZETA(I,NDX)+PZETA(I,NDX)
0012    100     CONTINUE
0013            IF(ANORM.NE.1.DO) THEN
        C MAKE SURE THE DENSITY INTEGRATES TO 1.0 EVEN IF ROUND-OFF OR
        C TRUNCATION ERROR.  THIS IS ESPECIALLY IMPORTANT
        C WHEN THE DENSITY FUNCTION HAS SINGULARITIES, AS IS THE
        C CASE FOR L1=1 OR L1=2 HOPS JAMMED.
0014              DO 200 I=1,10
0015              PZETA(I,NDX)=PZETA(I,NDX)/ANORM
0016    200       CONTINUE
0017            END IF
0018            RETURN
0019            END
```



```
0001            DOUBLE PRECISION FUNCTION PDFZET(ZETA,L1)
        C
        C DENSITY FUNCTION OF SUM OF COSINES
        C
        C   METHOD:
        C
        C       FOR L1=1 AND L1=2 HOPS, USE ANALYTICAL EXPRESSIONS
        C          FOR THE DENSITY FUNCTION
        C       FOR L1 .GE. 3 HOPS, USE INVERSE FOURIER TRANSFORM OF
        C          THE CHARACTERISTIC FUNCTION.  THE INVERSE FOURIER
        C          TRANSFORM IS COMPUTED BY GAUSSIAN QUADRATURE
        C          INTEGRATION WITH INTERVALS STEPPED BY 2*PI.
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            INTEGER*4 L1
0004            EXTERNAL CHARFZ
        C COMMON /NCOSES/ PASSES PARAMETERS TO INTEGRAND FUNCTION
0005            COMMON /NCOSES/ NCOS,Z
0006            DATA PI/3.141592653589793238462600/,PISQ/9.869604401089358600/
0007            IF(L1.EQ.1) THEN
        C  ALGEBRAIC FORM FOR DENSITY WHEN L1=1
0008              PDFZET=1.DO/(PI*DSQRT(1.DO-ZETA*ZETA))
0009            ELSE IF(L1.EQ.2) THEN
        C  WHEN L1.EQ.2, THE DENSITY IS AN ELLIPTIC INTEGRAL
0010              CALL DCEL1(ELIPIN,DSQRT(1.DO-0.25DO*ZETA*ZETA),KODE)
0011              IF(KODE.NE.0) THEN
0012                WRITE(5,1)KODE
0013    1           FORMAT(' DCEL1 KODE = ',I2)
0014                STOP
0015              END IF
0016              PDFZET=ELIPIN/PISQ
0017            ELSE
        C WHEN L1.GE.3, INVERSE TRANSFORM OF CHARACTERISTIC FUNCTION
0018              NCOS=L1
0019              Z=ZETA
0020              SUM=0.DO
0021              XL=0.DO
0022              DO 100 INTERV=1,120
0023              XU=XL+PI+PI
0024              CALL DGAU20(XL,XU,CHARFZ,PART)
0025              SUM=SUM+PART
0026              IF(DABS(PART).LE.1.D-6*DABS(SUM)) GOTO 200
0027              XL=XU
0028    100       CONTINUE
0029              WRITE(5,101)L1,ZETA,PART,SUM
0030    101       FORMAT(' L1=',I2,', ZETA=',1PD15.8,
             $ ', PZ(ZETA) NOT EVALUATED TO 6 PLACES.'/' PART=',1PD15.8,
             $ ' SUM=',D15.8)
0031    200       PDFZET=SUM/PI
0032            END IF
0033            RETURN
0034            END
```

```
0001            DOUBLE PRECISION FUNCTION CHARFZ(X)
          C
          C INTEGRAND FUNCTION FOR EVALUATION OF DENSITY OF SUM OF
          C COSINES OF UNIFORMLY DISTRIBUTED PHASE DIFFERENCES
          C BY INVERSE FOURIER TRANSFORM OF THE CHARACTERISTIC FUNCTION
          C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            COMMON /NCOSES/ NCOS, Z
0004            CALL DBESJ(X,0,BESSEL,1.D-6,KODE)
0005            IF(KODE.NE.0)WRITE(5,1)KODE,X
0006      1     FORMAT(' CHARFZ, BESJ KODE=',I2,' X=',1PD16.9)
0007            CHARFZ=DCOS(Z*X)*DXI(BESSEL,NCOS)
0008            RETURN
0009            END
```



```
0001            SUBROUTINE GENPIE(LL,MM,NTONES,NSTEP,NSLOTS,GOOD)
          C
          C SUBROUTINE TO GENERATE EVENT PROBABILITIES
          C FOR EVENLY SPACED TONES
          C
          C PROGRAMMER: A. KADRICHU
          C DATE: JAN 18, 1984
          C CONVERTED TO SUBROUTINE 11 APR 84 - R. H. FRENCH
          C
0002            IMPLICIT INTEGER*4(I-N),DOUBLE PRECISION (A-H,O-Z)
0003            LOGICAL*1 GO,GO2,STORE,NONE,GOOD,TFULL
0004            INTEGER*2 MM,LL,NSLOTS,NSTEP,NTONES
0005            DIMENSION LUP2(8),LUP3(8)
0006            DIMENSION IUPA(8)
0007            DIMENSION IUPD(8)
0008            DIMENSION LUP1(8)
          C **** WARNING ****
          C     COMMON BLOCK /VALIDS/ IS SHARED BY THE SV(20,105) ARRAY
          C     AND ARRAYS A(100) AND IASUB(100) TO SAVE ADDRESS SPACE
          C
          C     GENPIE IS CALLED ONLY FROM THE MAIN PROGRAM, AND NEEDS
          C     A(100) AND IASUB(100) ONLY AS WORK ARRAYS TO GENERATE
          C     FINAL RESULTS IN ARRAY D(200) IN COMMON BLOCK /EVENTS/.
          C     PSUBE IS CALLED AFTER GENPIE IS DONE, SO IT CAN SAFELY
          C     RE-USE THE STORAGE AS THE ARRAY OF VALIDITY INDICATORS
          C     FOR THE SAVED PDF SAMPLES FOR NUMERICAL INTEGRATION.
          C     PSUBE WILL BE DONE FOR ANY ONE CASE BEFORE
          C     GENPIE NEEDS THE WORK SPACE AGAIN FOR THE NEXT CASE.
          C
          C     THIS IS ADMITTEDLY POOR PROGRAMMING PRACTICE, BUT THE
          C     ADDRESS SPACE IS NEEDED TO FREE UP AN APR TO MAP TO THE
          C     VIRTUAL ARRAY SPDF WHICH IS USED TO SAVE UP THE PROBA-
          C     BILITIES FOR POSSIBLE RE-USE TO SAVE A LOT OF TIME.
          C
          C     ON SYSTEMS WITH MORE THAN 64 KBYTES DIRECTLY ADDRESSABLE,
          C     THIS SHARING WILL NOT BE NEEDED AND THE COMMON STATEMENT
          C     MAY BE REMOVED FROM SUBROUTINE GENPIE.
          C
          C  SIZE CONSIDERATIONS:
          C         DOUBLE PRECISION A(100) -->  100 X 8 = 800 BYTES
          C         INTEGER*4 IASUB(100)    -->  100 X 4 = 400 BYTES
          C         INTEGER*4 I(8)          -->    8 X 4 =  32 BYTES
          C         INTEGER*4 II(8)         -->    8 X 4 =  32 BYTES
          C         INTEGER*4 III(8)        -->    8 X 4 =  32 BYTES
          C         GO,GO2,STORE,NONE @ 1 BYTE EACH          4 BYTES
          C         N,M,NS,NQ,NUSEA,NUSEC @ 4 BYTES EA.     24 BYTES
          C         DM,DN,DNS,Q @ 8 BYTES EACH              32 BYTES
          C                                               ----------
          C         TOTAL SIZE                                  1356 BYTES
          C         FREE FOR FURTHER USE                         744 BYTES
0009            COMMON /VALIDS/ A(100),IASUB(100),I(8),II(8),III(8),
               $ GO,GO2,STORE,NONE,N,M,NS,NQ,DM,DN,DNS,Q,NUSEA,NUSEC
```

```
C C AND ICSUB SHARE STORAGE WITH CONDITIONAL ERROR
C PROBABILITIES IN PSUBE, AGAIN TO FREE UP ADDRESS SPACE
        COMMON /SHARE/ C(200),ICSUB(200)
C RETURN RESULTS IN COMMON /EVENTS/  (READ BY EVENT)
        COMMON /EVENTS/ D(200),IDSUB(200),NUSED
C SHARED STORAGE FOR COMMONLY NEEDED CONSTANT ARRAYS
        COMMON /SHARE2/ LOW(8),LINC(8),I200
        DATA I100/100/
        DATA IUPA/8*1/
        DATA LUP1/8*1/
C STORE=.FALSE. => DON'T STORE ZERO ELEMENTS OF SPARSE ARRAY.
        STORE=.FALSE.
        GOOD=.TRUE.
        IF(MM.EQ.2.OR.MM.EQ.4.OR.MM.EQ.8) GOTO 9999
        WRITE(6,9997) MM
9997    FORMAT(' M=',I5,'  NO FORMULA FOR EVENT PROBABILITY')
        GOOD=.FALSE.
        RETURN
9999    N=NSLOTS
        M=MM
        L=LL
        NS=NSTEP
        NQ=NTONES
        DM=MM
        DN=NSLOTS
        DNS=NS
        Q=NQ
        MAXQ=(NSLOTS-1)/NSTEP+1
        IF(NQ.GT.MAXQ) THEN
          WRITE(6,77)
77        FORMAT(' TOO MANY TONES FOR THE SPECIFIED SPACING')
          GOOD=.FALSE.
          RETURN
        END IF
        NSPAN=NSTEP*(NQ-1)+1
        IF(NSPAN.GT.N-2*M .AND. NQ.NE.MAXQ) THEN
          WRITE(6,78)
78        FORMAT(' EDGE-EFFECTS REGION NOT IMPLEMENTED')
          GOOD=.FALSE.
          RETURN
        END IF
        TFULL=NQ.EQ.MAXQ
        DO 80 IJ=1,M
        IUPD(IJ)=L
80      CONTINUE
C JAMMING PATTERN W/NON-ZERO PROBABILITY ON PER-HOP BASIS
        NUSEA=0
C INITIALIZE VECTOR-INDEX LOOP
        CALL VLINIT(I,LOW,M)
90      CONTINUE
        CALL LOCN(M,LOW,IUPA,I,ISUB)
        AIN=0.0D0
```



```
        CALL PR1HOP(I,M,NS,DN,Q,AIN,TFULL)
        CALL PUTIN(AIN,A,IASUB,NUSEA,I100,ISUB,IERR,STORE)
        IF(IERR.NE.0)STOP 'GENPIE OUT OF ROOM IN A'
C ITERATE VECTOR-INDEX LOOP
        CALL VLITER(I,LOW,LUP1,LINC,M,GO)
        IF(GO) GOTO 90
C COPY A INTO D ON BASIS OF EQUIVALENT SUBSCRIPTS
        NUSED=0
C INITIALIZE VECTOR-INDEX LOOP
        CALL VLINIT(I,LOW,M)
99      CONTINUE
        CALL LOCN(M,LOW,IUPA,I,ISUB1)
        CALL LOCN(M,LOW,IUPD,I,ISUB2)
        CALL LOOKUP(AOUT,A,IASUB,NUSEA,I100,ISUB1,STORE,NONE)
        CALL PUTIN(AOUT,D,IDSUB,NUSED,I200,ISUB2,IERR,STORE)
C ITERATE VECTOR-INDEX LOOP
        CALL VLITER(I,LOW,LUP1,LINC,M,GO)
        IF(GO)GOTO 99
C IF ONE HOP, NO CONVOLUTIONS NEEDED
        IF(L.EQ.1)RETURN
C ... L-1 CONVOLUTIONS ARE NEEDED ...
        DO 9998 L1=1,L-1
C SET UP VECTOR-LOOP UPPER LIMITS FOR THIS CONVOLUTION
9       DO 125 NN=1,M
        LUP2(NN)=L1
125     LUP3(NN)=L1+1
        NUSEC=0
C INITIALIZE VECTOR-LOOP TO ACCESS ARRAY A
        CALL VLINIT(I,LOW,M)
98      CONTINUE
C INITIALIZE VECTOR-LOOP TO ACCESS ARRAY D
        CALL VLINIT(II,LOW,M)
97      CONTINUE
C LOOK UP ELEMENTS AND PERFORM ONE TERM OF THE CONVOLUTION
        CALL LOCN(M,LOW,IUPA,I,ISUB1)
        CALL LOCN(M,LOW,IUPD,II,ISUB2)
        DO 21 NN=1,M
        III(NN)=I(NN)+II(NN)
21      CONTINUE
        CALL LOCN(M,LOW,IUPD,III,ISUB3)
        CALL LOOKUP(AOUT,A,IASUB,NUSEA,I100,ISUB1,STORE,NONE)
        CALL LOOKUP(DOUT,D,IDSUB,NUSED,I200,ISUB2,STORE,NONE)
        CALL LOOKUP(COUT,C,ICSUB,NUSEC,I200,ISUB3,STORE,NONE)
        CIN=COUT+AOUT*DOUT
        CALL PUTIN(CIN,C,ICSUB,NUSEC,I200,ISUB3,IERR,STORE)
        IF(IERR.NE.0) STOP 'GENPIE OUT OF ROOM IN C'
C ITERATE VECTOR-LOOP FOR ARRAY D
        CALL VLITER(II,LOW,LUP2,LINC,M,GO2)
        IF(GO2) GOTO 97
C ITERATE VECTOR-LOOP FOR ARRAY A
        CALL VLITER(I,LOW,LUP1,LINC,M,GO)
        IF(GO) GOTO 98
```

```
      C COPY C TO D IN SORTED ORDER FOR NEXT ITERATION
0095          NUSED=0
0096          CALL VLINIT(II,LOW,M)
0097  96      CONTINUE
0098          CALL LOCN(M,LOW,IUPD,II,ISUB)
0099          CALL LOOKUP(COUT,C,ICSUB,NUSEC,I200,ISUB,STORE,NONE)
0100          DIN=COUT
0101          CALL PUTIN(DIN,D,IDSUB,NUSED,I200,ISUB,IERR,STORE)
0102          IF(IERR.NE.0) STOP 'GENPIE OUT OF ROOM IN D'
0103          CALL VLITER(II,LOW,LUP3,LINC,M,GO)
0104          IF(GO) GOTO 96
      C ITERATE THE CONVOLUTION
0105  9998    CONTINUE
0106          RETURN
0107          END
```



```
0001          SUBROUTINE PUTIN(CIN,C,ICSUB,NUSE,NMAX,K,IERR,STORE)
      C
      C THIS SUBROUTINE INSERTS AN ELEMENT INTO A SPARSE ARRAY FOR
      C WHICH ONLY THE NON-ZERO ELEMENTS ARE KEPT IN STORAGE IF
      C THE SWITCH STORE IS .FALSE.
      C
      C THE DOUBLE PRECISION VALUE CIN IS STORED AS C(K) WHERE
      C THE AUXILIARY ARRAY ICSUB IS USED TO KEEP TRACK OF THE
      C SUBSCRIPTS OF THE CORRESPONDING ENTRIES IN C.
      C LONG (4-BYTE) INTEGERS ARE USED TO ACCOMODATE LARGE
      C SUBSCRIPT VALUES FOR THE SPARSE ARRAY C.
      C
      C USAGE:
      C       LOGICAL*1 STORE
      C       INTEGER*4 ICSUB(NMAX),NUSE,NMAX,K,IERR
      C       DOUBLE PRECISION C(NMAX),CIN
      C       CALL PUTIN(CIN,C,ICSUB,NUSE,NMAX,K,IERR,STORE)
      C WHERE
      C  CIN = VALUE OF ELEMENT TO STORE
      C  C = ARRAY IN WHICH NON-ZERO VALUES ARE ACTUALLY STORED
      C  ICSUB = AUXILIARY ARRAY FOR ACTUAL SUBSCRIPT VALUES
      C  NUSE = NUMBER OF ELEMENTS OF C CURRENTLY OCCUPIED
      C  NMAX = SIZE OF ARRAY C
      C  IERR = ERROR RETURN CODE, 0 IF NO ERROR OR 1 IF THERE IS
      C      NO ROOM AVAILABLE IN C
      C  STORE = .TRUE. TO STORE ZEROES EXPLICITLY, ELSE .FALSE.
      C NOTE: IF CIN=0 AND THE SUBSCRIPT K IS FOUND IN ICSUB, THEN
      C       THE ELEMENT IS DELETED BY SHIFTING DOWNWARD ALL
      C       FOLLOWING ELEMENTS OF THE ARRAY
      C
      C PROGRAMMER: ROBERT H. FRENCH          DATE: 11 JANUARY 1984
      C
0002          IMPLICIT INTEGER*4(I-N),DOUBLE PRECISION(A-H,O-Z)
0003          INTEGER*4 K,ICSUB(NMAX)
0004          LOGICAL*1 STORE
0005          DIMENSION C(NMAX)
0006          IERR=0
0007          IF(STORE) GOTO 5
0008          IF(CIN.EQ.0.D0) GOTO30
0009  5       IF(NUSE.EQ.0)GOTO 20
0010          DO 10 I=1,NUSE
0011          IF(ICSUB(I).NE.K) GOTO 10
0012          C(I)=CIN
0013          RETURN
0014  10      CONTINUE
0015          IF(NUSE.LT.NMAX) GOTO 20
0016          IERR=1
0017          RETURN
0018  20      NUSE=NUSE+1
0019          ICSUB(NUSE)=K
0020          C(NUSE)=CIN
0021          RETURN
```

```
0022   30      DO 40 I=1,NUSE
0023           J=I
0024           IF(ICSUB(I).EQ.K) GOTO 50
0025   40      CONTINUE
0026           RETURN
       C
       C REMOVE THE ZEROED ELEMENT AND BUMP COUNT OF ENTRIES USED
       C
0027   50      DO 60 I=J,NUSE-1
0028           ICSUB(I)=ICSUB(I+1)
0029           C(I)=C(I+1)
0030   60      CONTINUE
0031           NUSE=NUSE-1
0032           RETURN
0033           END
```



```
0001           SUBROUTINE LOOKUP(COUT,C,ICSUB,N,NMAX,K,STORE,NONE)
       C
       C THIS SUBROUTINE RETRIEVES AN ELEMENT OF A SPARSE ARRAY WHICH
       C HAS BEEN STORED COMPACTLY
       C
       C THE ARRAY IS DOUBLE PRECISION.  ALL INTEGERS ARE 4-BYTES
       C TO ALLOW FOR LARGE VALUES OF THE SUBSCRIPT.
       C
       C USAGE:
       C       INTEGER*4 ICSUB(NMAX),N,NMAX,K
       C       LOGICAL*1 STORE, NONE
       C       DOUBLE PRECISION C(NMAX),C
       C       CALL LOOKUP(COUT,C,ICSUB,N,NMAX,K,STORE,NONE)
       C WHERE
       C         COUT = VALUE OF C(K) (OUTPUT FROM SUBROUTINE)
       C            C = ARRAY USED TO STORE NON-ZERO ELEMENTS
       C        ICSUB = AUXILIARY ARRAY TO STORE ACTUAL SUBSCRIPTS
       C            N = NUMBER OF ELEMENTS OF C CURRENTLY IN USE
       C         NMAX = SIZE OF C
       C            K = SUBSCRIPT OF SPARSE ARRAY TO LOOK UP
       C        STORE = .TRUE. IF ZEROES STORED EXPLICITLY, ELSE .FALSE.
       C         NONE = .FALSE. IF ZEROES NOT STORED OR ZEROES STORED AND
       C                        ELEMENT IS FOUND IN THE STORED ARRAY
       C                .TRUE.  IF ZEROES ARE STORED AND THE ELEMENT IS
       C                NOT FOUND (OUTPUT QUANTITY)
       C
       C PROGRAMMER: ROBERT H. FRENCH
       C       DATE: 11 JANUARY 1984
       C
0002           IMPLICIT INTEGER*4(I-N),DOUBLE PRECISION(A-H,O-Z)
0003           INTEGER*4 ICSUB(NMAX),K
0004           LOGICAL*1 STORE, NONE
0005           DIMENSION C(NMAX)
0006           NONE=.FALSE.
0007           DO 10 I=1,N
0008           IF(ICSUB(I).NE.K)GOTO 10
0009           COUT=C(I)
0010           RETURN
0011   10      CONTINUE
0012           IF(STORE) THEN
0013             NONE=.TRUE.
0014           ELSE
0015             COUT=0.
0016           END IF
0017           RETURN
0018           END
```

```
PDP-11 FORTRAN-77 V4.0-1        10:58:10      3-MAY-84           Page 26

0001          SUBROUTINE LOCN(NDIM,ILOW,IUP,ISUB,LINEAR)
        C
        C THIS SUBROUTINE COMPUTES THE EQUIVALENT LINEAR SUBSCRIPT FOR
        C A MULTIDIMENSIONAL ARRAY OF NDIM DIMENSIONS
        C
        C IF THE ARRAY A IS DEFINED AS
        C       DIMENSION A(ILOW(1):IUP(1),....,ILOW(NDIM):IUP(NDIM))
        C AND ISUB(1),....,ISUB(NDIM) IS A SET OF SUBSCRIPTS FOR A,
        C THEN THIS SUBROUTINE RETURNS IN LINEAR THE OFFSET FROM THE
        C ORIGIN OF A TO THE ELEMENT A(ISUB(1),....,ISUB(NDIM)), ASSUMING
        C THE FIRST SUBSCRIPT VARIES MOST RAPIDLY.
        C
        C USAGE:
        C       DIMENSION ILOW(NDIM),IUP(NDIM),ISUB(NDIM)
        C       DATA ILOW/lower limits of defined subscripts of array/
        C       DATA IUP/upper limits of defined subscripts of array/
        C       ...SET ISUB TO DESIRED SEQUENCE OF SUBSCRIPTS...
        C       CALL LOCN(NDIM,ILOW,IUP,ISUB,LINEAR)
        C WHERE
        C       NDIM = NUMBER OF DIMENSIONS THE ARRAY HAS
        C       ILOW = ARRAY OF LOWER SUBSCRIPT BOUNDS
        C        IUP = ARRAY OF UPPER SUBSCRIPT BOUNDS
        C       ISUB = ARRAY CONTAINING SUBSCRIPT FOR WHICH LOCATION IS
        C              TO BE COMPUTED
        C     LINEAR = RETURNED VALUE OF OFFSET INTO ARRAY IN MEMORY
        C
        C NOTE: ALL INTEGERS ARE 4-BYTE INTEGERS TO ALLOW FOR LARGE
        C       SUBSCRIPT VALUES
        C
        C PROGRAMMER: ROBERT H. FRENCH
        C       DATE: 11 JANUARY 1984
        C
0002          IMPLICIT INTEGER*4(I-N)
0003          DIMENSION ILOW(NDIM),IUP(NDIM),ISUB(NDIM)
0004          LINEAR=0
0005          DO 10 I=1,NDIM-1
0006          J=NDIM-I+1
0007          LINEAR=(LINEAR+(ISUB(J)-ILOW(J)))*(IUP(J-1)-ILOW(J-1)+1)
0008    10    CONTINUE
0009          LINEAR=LINEAR+ISUB(1)-ILOW(1)
0010          RETURN
0011          END
```

```
PDP-11 FORTRAN-77 V4.0-1        10:58:14      3-MAY-84           Page 27

0001          SUBROUTINE VLINIT(LVEC,LLOW,LMAX)
        C
        C THIS SUBROUTINE INITIALIZES A "VECTOR DO-LOOP" STRUCTURE
        C DEFINED BY THE FOLLOWING PSEUDO-FORTRAN CODE:
        C       DO 100 LVEC(1)=LLOW(1),LUP(1),LINC(1)
        C       DO 100 LVEC(2)=LLOW(2),LUP(2),LINC(2)
        C                     :
        C                     :
        C       DO 100 LVEC(LMAX)=LLOW(LMAX),LUP(LMAX),LINC(LMAX)
        C                     :
        C                     :
        C       (STATEMENTS IN RANGE OF LOOP)
        C 100   CONTINUE
        C
        C THE COMPANION ROUTINE VLITER HANDLES THE LOOP CONTROL AT THE
        C CONTINUE STATEMENT IN THE ABOVE STRUCTURE
        C
        C USAGE:
        C       LOGICAL*1 GO
        C       DIMENSION LVEC(LMAX),LLOW(LMAX),LUP(LMAX),LINC(LMAX)
        C (INITIALIZE ARRAY LLOW TO STARTING VALUES OF THE NESTED LOOPS)
        C (INITIALIZE ARRAY LUP TO STOPING VALUES OF THE NESTED LOOPS)
        C (INITIALIZE ARRAY LINC TO INCREMENTS OF THE LOOPS)
        C       CALL VLINIT(LVEC,LLOW,LMAX)
        C   100 CONTINUE
        C         .
        C         . (STATEMENTS IN RANGE OF LOOPS)
        C         .
        C       CALL VLITER(LVEC,LLOW,LUP,LINC,LMAX,GO)
        C       IF(GO)GOTO 100
        C WHERE
        C    LVEC = ARRAY FOR STORAGE OF LOOP INDICES. LVEC(1) IS THE
        C           OUTER-MOST LOOP; LVEC(LMAX), THE INNER-MOST LOOP.
        C    LLOW = ARRAY FOR STORAGE OF LOOP STARTING VALUES, IN SAME
        C           SEQUENCE AS LVEC
        C    LUP = ARRAY FOR STORAGE OF LOOP FINAL VALUES, IN SAME
        C          SEQUENCE AS LVEC
        C    LINC = ARRAY FOR STORAGE OF LOOP INCREMENTS, IN SAME
        C           SEQUENCE AS LVEC
        C    LMAX = NUMBER OF LOOPS NESTED
        C    GO = LOGICAL VARIABLE, .TRUE. IF JUMP BACK TO BEGINNING OF
        C         STATEMENTS IN THE RANGE OF THE LOOP SHOULD OCCUR,
        C         .FALSE. OTHERWISE (I.E. OUTER-MOST LOOP TERMINATED)
        C
        C PROGRAMMER: ROBERT H. FRENCH         DATE: 11 JANUARY 1984
0002          IMPLICIT INTEGER*4(I-N)
0003          DIMENSION LVEC(LMAX),LLOW(LMAX)
0004          DO 1 N=1,LMAX
0005          LVEC(N)=LLOW(N)
0006    1     CONTINUE
0007          RETURN
0008          END
```

```
      SUBROUTINE VLITER(LVEC,LLOW,LUP,LINC,LMAX,GO)
C
C LOOP ITERATION LOGIC FOR A "VECTOR DO-LOOP"
C
C SEE DETAILED COMMENTS IN SUBROUTINE VLINIT FOR USAGE AND
C PARAMETER DEFINITIONS
C
C PROGRAMMER: ROBERT H. FRENCH
C       DATE: 11 JANUARY 1984
C
      IMPLICIT INTEGER*4(I-N)
      LOGICAL*1 GO
      DIMENSION LVEC(LMAX),LLOW(LMAX),LUP(LMAX),LINC(LMAX)
      GO=.TRUE.
      DO 100 NDX=1,LMAX
      NSUB=LMAX+1-NDX
      LVEC(NSUB)=LVEC(NSUB)+LINC(NSUB)
      IF((LINC(NSUB).GE.0.AND.LVEC(NSUB).LE.LUP(NSUB))
     $  .OR.(LINC(NSUB).LT.0.AND.LVEC(NSUB).GE.LUP(NSUB))) RETURN
      LVEC(NSUB)=LLOW(NSUB)
100   CONTINUE
      GO=.FALSE.
      RETURN
      END
```



```
      SUBROUTINE PR1HOP(I,M,NS,DN,Q,AIN,TFULL)
C
C THIS SUBROUTINE COMPUTES THE EVENT PROBABILITY FOR ALL
C POSSIBLE JAMMING PATTERNS WITH NON-ZERO PROBABILITY FOR
C L=1 HOP/SYMBOL FOR EVENLY SPACED TONES.
C
C THE DISTANCE BETWEEN TWO ADJACENT TONES IS EQUAL TO A
C MULTIPLE (n) OF THE HOPPING RATE.
C
C PROGRAMMER: A. KADRICHU      DATE: JAN 19, 1984
C NAME CHANGED TO PR1HOP - R.H. FRENCH, 11 APRIL 1984
C LOGICAL VARIABLES USED TO CUT TASK SIZE (SAVES ABOUT 2.5 KB
C OVER VERSION IN APPENDIX 8C OF REPORT)--RHF, 16 APR 84
C
C FULL-BAND (MAXIMUM Q) CASE ADDED - RHF, 2 MAY 84
C THE FULL-BAND CASE IS AN APPROXIMATION, IN THAT IT IS
C ASSUMED THAT AVERAGING OVER ALL POSSIBLE POSITIONS WITH
C RESPECT TO THE BAND EDGE RESULTS IN THOSE POSSIBLE
C EVENTS HAVING EQUAL PROBABILITIES = 1/(NO. OF POSSIBLE EVENTS)
C
      IMPLICIT INTEGER*4(I-N),DOUBLE PRECISION (A-H,O-S,U-Z)
      IMPLICIT LOGICAL*1 (T)
      DIMENSION I(8)
      DM=M
      DNS=NS
      DN1=DN-1.D0
      DN3=DN-3.D0
      DN7=DN-7.D0
      AIN=0.D0
C
C FOR M=2 WITH TONE SPACED AT 1 AND 2 B HZ APART
C
      IF(M.EQ.2) THEN
        T1=I(1).EQ.0
        T2=I(2).EQ.0
        IF(NS.EQ.1) THEN
         IF(TFULL) THEN
          IF(.NOT.T1.AND..NOT.T2) AIN=1.D0
         ELSE
          IF(T1.AND.T2) AIN=(DN-Q-2.D0)/DN1
          IF(T1.XOR.T2) AIN=1.D0/DN1
          IF(.NOT.T1.AND..NOT.T2) AIN=(Q-1.D0)/DN1
         END IF
C
        ELSE IF(NS.GE.2) THEN
         IF(TFULL.AND.NS.EQ.2) THEN
          IF(T1.XOR.T2) AIN=0.5D0
         ELSE
          IF(T1.AND.T2) AIN=(DN-2.D0*Q-1.D0)/DN1
          IF(T1.XOR.T2) AIN=Q/DN1
         END IF
        END IF
```

```
C
C FOR M=4 WITH TONE SPACED AT 1,2,3, AND 4 B HZ APART
C
      ELSE IF(M.EQ.4) THEN
        T1=I(1).EQ.0
        T2=I(2).EQ.0
        T3=I(3).EQ.0
        T4=I(4).EQ.0
        IF(NS.EQ.1) THEN
         IF(TFULL) THEN
          IF(.NOT.T1.AND..NOT.T2.AND..NOT.T3.AND..NOT.T4)
     $      AIN=1.D0
         ELSE
          IF(T1.AND.T2.AND.T3.AND.T4) AIN=(DN-Q-6.D0)/DN3
          IF(T1.AND.T2.AND.T3.AND..NOT.T4) AIN=1.D0/DN3
          IF(T1.AND.T2.AND..NOT.T3.AND..NOT.T4) AIN=1.D0/DN3
          IF(T1.AND..NOT.T2.AND..NOT.T3.AND..NOT.T4) AIN=1.D0/DN3
          IF(.NOT.T1.AND.T2.AND.T3.AND.T4) AIN=1.D0/DN3
          IF(.NOT.T1.AND..NOT.T2.AND.T3.AND.T4) AIN=1.D0/DN3
          IF(.NOT.T1.AND..NOT.T2.AND..NOT.T3.AND.T4) AIN=1.D0/DN3
          IF(.NOT.T1.AND..NOT.T2.AND..NOT.T3.AND..NOT.T4)
     $      AIN=(Q-3.D0)/DN3
         END IF
C
        ELSE IF(NS.EQ.2) THEN
         IF(TFULL) THEN
          IF(T1.AND..NOT.T2.AND.T3.AND..NOT.T4) AIN=0.5D0
          IF(.NOT.T1.AND.T2.AND..NOT.T3.AND.T4) AIN=0.5D0
         ELSE
          IF(T1.AND.T2.AND.T3.AND.T4) AIN=(DN-2.D0*Q-5.D0)/DN3
          IF(T1.AND.T2.AND.T3.AND..NOT.T4) AIN=1.D0/DN3
          IF(T1.AND.T2.AND..NOT.T3.AND.T4) AIN=1.D0/DN3
          IF(T1.AND..NOT.T2.AND.T3.AND.T4) AIN=1.D0/DN3
          IF(T1.AND..NOT.T2.AND.T3.AND..NOT.T4) AIN=(Q-1.D0)/DN3
          IF(.NOT.T1.AND.T2.AND.T3.AND.T4) AIN=1.D0/DN3
          IF(.NOT.T1.AND.T2.AND..NOT.T3.AND.T4) AIN=(Q-1.D0)/DN3
         END IF
C
        ELSE IF(NS.EQ.3) THEN
         IF(TFULL) THEN
          IF(.NOT.T1.AND.T2.AND.T3.AND..NOT.T4)
     $      AIN=0.3333333333333333D0
          IF(T1.AND.T2.AND..NOT.T3.AND.T4) AIN=0.3333333333333333D0
          IF(T1.AND..NOT.T2.AND.T3.AND.T4) AIN=0.3333333333333333D0
         ELSE
          IF(T1.AND.T2.AND.T3.AND.T4) AIN=(DN-3.D0*Q-4.D0)/DN3
          IF(T1.AND.T2.AND.T3.AND..NOT.T4) AIN=1.D0/DN3
          IF(T1.AND.T2.AND..NOT.T3.AND.T4) AIN=Q/DN3
          IF(T1.AND..NOT.T2.AND.T3.AND.T4) AIN=Q/DN3
          IF(.NOT.T1.AND.T2.AND.T3.AND.T4) AIN=1.D0/DN3
          IF(.NOT.T1.AND.T2.AND.T3.AND..NOT.T4) AIN=(Q-1.D0)/DN3
         END IF
```



```
C
        ELSE IF(NS.GE.4) THEN
         IF(TFULL.AND.NS.EQ.4) THEN
          IF(T1.AND.T2.AND.T3.AND..NOT.T4) AIN=0.25D0
          IF(T1.AND.T2.AND..NOT.T3.AND.T4) AIN=0.25D0
          IF(T1.AND..NOT.T2.AND.T3.AND.T4) AIN=0.25D0
          IF(.NOT.T1.AND.T2.AND.T3.AND.T4) AIN=0.25D0
         ELSE
          IF(T1.AND.T2.AND.T3.AND.T4) AIN=(DN-4.D0*Q-3.D0)/DN3
          IF(T1.AND.T2.AND.T3.AND..NOT.T4) AIN=Q/DN3
          IF(T1.AND.T2.AND..NOT.T3.AND.T4) AIN=Q/DN3
          IF(T1.AND..NOT.T2.AND.T3.AND.T4) AIN=Q/DN3
          IF(.NOT.T1.AND.T2.AND.T3.AND.T4) AIN=Q/DN3
         END IF
        END IF
C
C FOR M=8 WITH TONE SPACED AT 1 UP TO 8 B HZ APART
C
      ELSE IF(M.EQ.8) THEN
        T2=I(2).EQ.0
        T3=I(3).EQ.0
        T4=I(4).EQ.0
        T5=I(5).EQ.0
        T6=I(6).EQ.0
        T7=I(7).EQ.0
        T8=I(8).EQ.0
        IF(NS.EQ.1) THEN
         IF(TFULL) THEN
          IF(.NOT.T1.AND..NOT.T2.AND..NOT.T3.AND.
     $       .NOT.T4.AND..NOT.T5.AND..NOT.T6.AND.
     $       .NOT.T7.AND..NOT.T8) AIN=1.D0
         ELSE
          IF(T1.AND.T2.AND.T3.AND.T4
     $    .AND.T5.AND.T6.AND.T7.AND.T8) AIN=(DN-Q-14.D0)/DN7
          IF(T1.AND.T2.AND.T3.AND.T4
     $    .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=1.D0/DN7
          IF(T1.AND.T2.AND.T3.AND.T4
     $    .AND.T5.AND.T6.AND..NOT.T7.AND..NOT.T8) AIN=1.D0/DN7
          IF(T1.AND.T2.AND.T3.AND.T4
     $    .AND.T5.AND..NOT.T6.AND..NOT.T7.AND..NOT.T8) AIN=1.D0/DN7
          IF(T1.AND.T2.AND.T3.AND.T4.AND..NOT.T5
     $    .AND..NOT.T6.AND..NOT.T7.AND..NOT.T8) AIN=1.D0/DN7
          IF(T1.AND.T2.AND.T3.AND..NOT.T4.AND..NOT.T5
     $    .AND..NOT.T6.AND..NOT.T7.AND..NOT.T8) AIN=1.D0/DN7
          IF(T1.AND.T2.AND..NOT.T3.AND..NOT.T4.AND..NOT.T5
     $    .AND..NOT.T6.AND..NOT.T7.AND..NOT.T8) AIN=1.D0/DN7
          IF(T1.AND..NOT.T2.AND..NOT.T3.AND..NOT.T4
     $    .AND..NOT.T5.AND..NOT.T6.AND..NOT.T7.AND..NOT.T8)
     $    AIN=1.D0/DN7
          IF(.NOT.T1.AND.T2.AND.T3.AND.T4
     $    .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.D0/DN7
```

```
PDP-11 FORTRAN-77 V4.0-1        10:58:22     3-MAY-84           Page 32

0110            IF(.NOT.T1.AND..NOT.T2.AND.T3.AND.T4
        $       .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0111            IF(.NOT.T1.AND..NOT.T2.AND..NOT.T3.AND.T4
        $       .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0112            IF(.NOT.T1.AND..NOT.T2.AND..NOT.T3.AND..NOT.T4
        $       .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0113            IF(.NOT.T1.AND..NOT.T2.AND..NOT.T3.AND..NOT.T4
        $       .AND..NOT.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0114            IF(.NOT.T1.AND..NOT.T2.AND..NOT.T3.AND..NOT.T4
        $       .AND..NOT.T5.AND..NOT.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0115            IF(.NOT.T1.AND..NOT.T2.AND..NOT.T3.AND..NOT.T4.AND.
        $       .NOT.T5.AND..NOT.T6.AND..NOT.T7.AND.T8) AIN=1.DO/DN7
0116            IF(.NOT.T1.AND..NOT.T2.AND..NOT.T3.AND..NOT.T4
        $       .AND..NOT.T5.AND..NOT.T6.AND..NOT.T7.AND..NOT.T8)
        $       AIN=(Q-7.DO)/DN7
0117           END IF
     C
0118          ELSE IF(NS.EQ.2) THEN
0119           IF(TFULL) THEN
0120            IF(T1.AND..NOT.T2.AND.T3.AND..NOT.T4
        $       .AND.T5.AND..NOT.T6.AND.T7.AND..NOT.T8) AIN=0.5DO
0121            IF(.NOT.T1.AND.T2.AND..NOT.T3.AND.T4
        $       .AND..NOT.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=0.5DO
0122           ELSE
0123            IF(T1.AND.T2.AND.T3.AND.T4.AND.T5
        $       .AND.T6.AND.T7.AND.T8) AIN=(DN-2.DO*Q-13.DO)/DN7
0124            IF(T1.AND.T2.AND.T3.AND.T4
        $       .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=1.DO/DN7
0125            IF(T1.AND.T2.AND.T3.AND.T4
        $       .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=1.DO/DN7
0126            IF(T1.AND.T2.AND.T3.AND.T4.AND.T5
        $       .AND..NOT.T6.AND.T7.AND..NOT.T8) AIN=1.DO/DN7
0127            IF(T1.AND.T2.AND.T3.AND.T4.AND..NOT.T5
        $       .AND.T6.AND..NOT.T7.AND.T8)AIN=1.DO/DN7
0128            IF(T1.AND.T2.AND.T3.AND..NOT.T4.AND.T5
        $       .AND..NOT.T6.AND.T7.AND..NOT.T8) AIN=1.DO/DN7
0129            IF(T1.AND.T2.AND..NOT.T3.AND.T4.AND..NOT.T5
        $       .AND.T6.AND..NOT.T7.AND.T8) AIN=1.DO/DN7
0130            IF(T1.AND..NOT.T2.AND.T3.AND.T4.AND.T5
        $       .AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0131            IF(T1.AND..NOT.T2.AND.T3.AND..NOT.T4
        $       .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0132            IF(T1.AND..NOT.T2.AND.T3.AND..NOT.T4
        $       .AND.T5.AND..NOT.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0133            IF(T1.AND..NOT.T2.AND.T3.AND..NOT.T4.AND.T5
        $       .AND..NOT.T6.AND.T7.AND..NOT.T8) AIN=(Q-3.DO)/DN7
0134            IF(.NOT.T1.AND.T2.AND.T3.AND.T4
        $       .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0135            IF(.NOT.T1.AND.T2.AND..NOT.T3.AND.T4
        $       .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0136            IF(.NOT.T1.AND.T2.AND..NOT.T3.AND.T4
        $       .AND..NOT.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
```

```
PDP-11 FORTRAN-77 V4.0-1        10:58:22     3-MAY-84           Page 33

0137            IF(.NOT.T1.AND.T2.AND..NOT.T3.AND.T4.AND..NOT.T5
        $       .AND.T6.AND..NOT.T7.AND.T8) AIN=(Q-3.DO)/DN7
0138           END IF
     C
0139          ELSE IF(NS.EQ.3) THEN
0140           IF(TFULL) THEN
0141            IF(.NOT.T1.AND.T2.AND.T3.AND..NOT.T4.AND.T5
        $       .AND.T6.AND..NOT.T7.AND.T8) AIN=0.3333333333333333DO
0142            IF(T1.AND..NOT.T2.AND.T3.AND.T4.AND..NOT.T5
        $       .AND.T6.AND.T7.AND..NOT.T8) AIN=0.3333333333333333DO
0143            IF(T1.AND.T2.AND..NOT.T3.AND.T4.AND.
        $       T5.AND..NOT.T6.AND.T7.AND.T8) AIN=0.3333333333333333DO
0144           ELSE
0145            IF(T1.AND.T2.AND.T3.AND.T4.AND.T5
        $       .AND.T6.AND.T7.AND.T8) AIN=(DN-3.DO*Q-12.DO)/DN7
0146            IF(T1.AND.T2.AND.T3.AND.T4
        $       .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=1.DO/DN7
0147            IF(T1.AND.T2.AND.T3.AND.T4
        $       .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=1.DO/DN7
0148            IF(T1.AND.T2.AND.T3.AND.T4
        $       .AND.T5.AND..NOT.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0149            IF(T1.AND.T2.AND.T3.AND.T4
        $       .AND..NOT.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=1.DO/DN7
0150            IF(T1.AND.T2.AND.T3.AND..NOT.T4
        $       .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=1.DO/DN7
0151            IF(T1.AND.T2.AND..NOT.T3.AND.T4
        $       .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0152            IF(T1.AND.T2.AND..NOT.T3.AND.T4
        $       .AND.T5.AND..NOT.T6.AND.T7.AND.T8) AIN=(Q-1.DO)/DN7
0153            IF(T1.AND..NOT.T2.AND.T3.AND.T4
        $       .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0154            IF(T1.AND..NOT.T2.AND.T3.AND.T4
        $       .AND..NOT.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0155            IF(T1.AND..NOT.T2.AND.T3.AND.T4.AND..NOT.T5
        $       .AND.T6.AND.T7.AND..NOT.T8) AIN=(Q-2.DO)/DN7
0156            IF(.NOT.T1.AND.T2.AND.T3.AND.T4
        $       .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0157            IF(.NOT.T1.AND.T2.AND.T3.AND..NOT.T4
        $       .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.DO/DN7
0158            IF(.NOT.T1.AND.T2.AND.T3.AND..NOT.T4
        $       .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=(Q-2.DO)/DN7
0159           END IF
     C
0160          ELSE IF(NS.EQ.4) THEN
0161           IF(TFULL) THEN
0162            IF(.NOT.T1.AND.T2.AND.T3.AND.T4.AND.
        $       .NOT.T5.AND.T6.AND.T7.AND.T8) AIN=0.25DO
0163            IF(T1.AND..NOT.T2.AND.T3.AND.T4.AND.
        $       T5.AND..NOT.T6.AND.T7.AND.T8) AIN=0.25DO
0164            IF(T1.AND.T2.AND..NOT.T3.AND.T4.AND.
        $       T5.AND.T6.AND..NOT.T7.AND.T8) AIN=0.25DO
0165            IF(T1.AND.T2.AND.T3.AND..NOT.T4.AND.
        $       T5.AND.T6.AND.T7.AND..NOT.T8) AIN=0.25DO
```

```
0166            ELSE
0167             IF(T1.AND.T2.AND.T3.AND.T4.AND.T5
          $      .AND.T6.AND.T7.AND.T8) AIN=(DN-4.D0*Q-11.D0)/DN7
0168             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=1.D0/DN7
0169             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=1.D0/DN7
0170             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND..NOT.T6.AND.T7.AND.T8) AIN=1.D0/DN7
0171             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND..NOT.T5.AND.T6.AND.T7.AND.T8) AIN=1.D0/DN7
0172             IF(T1.AND.T2.AND.T3.AND..NOT.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.D0/DN7
0173             IF(T1.AND.T2.AND.T3.AND..NOT.T4
          $      .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=(Q-1.D0)/DN7
0174             IF(T1.AND.T2.AND..NOT.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.D0/DN7
0175             IF(T1.AND.T2.AND..NOT.T3.AND.T4
          $      .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=(Q-1.D0)/DN7
0176             IF(T1.AND..NOT.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.D0/DN7
0177             IF(T1.AND..NOT.T2.AND.T3.AND.T4
          $      .AND.T5.AND..NOT.T6.AND.T7.AND.T8) AIN=(Q-1.D0)/DN7
0178             IF(.NOT.T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.D0/DN7
0179             IF(.NOT.T1.AND.T2.AND.T3.AND.T4
          $      .AND..NOT.T5.AND.T6.AND.T7.AND.T8) AIN=(Q-1.D0)/DN7
0180            END IF
      C
0181           ELSE IF(NS.EQ.5) THEN
0182            IF(TFULL) THEN
0183             IF(.NOT.T1.AND.T2.AND.T3.AND.T4.AND.
          $      T5.AND..NOT.T6.AND.T7.AND.T8) AIN=0.2D0
0184             IF(T1.AND..NOT.T2.AND.T3.AND.T4.AND.
          $      T5.AND.T6.AND..NOT.T7.AND.T8) AIN=0.2D0
0185             IF(T1.AND.T2.AND..NOT.T3.AND.T4.AND.
          $      T5.AND.T6.AND.T7.AND..NOT.T8) AIN=0.2D0
0186             IF(T1.AND.T2.AND.T3.AND..NOT.T4.AND.
          $      T5.AND.T6.AND.T7.AND.T8) AIN=0.2D0
0187             IF(T1.AND.T2.AND.T3.AND.T4.AND.
          $      .NOT.T5.AND.T6.AND.T7.AND.T8) AIN=0.2D0
0188            ELSE
0189             IF(T1.AND.T2.AND.T3.AND.T4.AND.T5
          $      .AND.T6.AND.T7.AND.T8) AIN=(DN-5.D0*Q-10.D0)/DN7
0190             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=1.D0/DN7
0191             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=1.D0/DN7
0192             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND..NOT.T6.AND.T7.AND.T8) AIN=1.D0/DN7
0193             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND..NOT.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
```



```
0194             IF(T1.AND.T2.AND.T3.AND..NOT.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0195             IF(T1.AND.T2.AND..NOT.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.D0/DN7
0196             IF(T1.AND.T2.AND..NOT.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=(Q-1.D0)/DN7
0197             IF(T1.AND..NOT.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.D0/DN7
0198             IF(T1.AND..NOT.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=(Q-1.D0)/DN7
0199             IF(.NOT.T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.D0/DN7
0200             IF(.NOT.T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND..NOT.T6.AND.T7.AND.T8) AIN=(Q-1.D0)/DN7
0201            END IF
      C
0202           ELSE IF(NS.EQ.6) THEN
0203            IF(TFULL) THEN
0204             IF(.NOT.T1.AND.T2.AND.T3.AND.T4.AND.
          $      T5.AND.T6.AND..NOT.T7.AND.T8) AIN=0.1666666666666667D0
0205             IF(T1.AND..NOT.T2.AND.T3.AND.T4.AND.
          $      T5.AND.T6.AND.T7.AND..NOT.T8) AIN=0.1666666666666667D0
0206             IF(T1.AND.T2.AND..NOT.T3.AND.T4.AND.
          $      T5.AND.T6.AND.T7.AND.T8) AIN=0.1666666666666667D0
0207             IF(T1.AND.T2.AND.T3.AND..NOT.T4.AND.
          $      T5.AND.T6.AND.T7.AND.T8) AIN=0.1666666666666667D0
0208             IF(T1.AND.T2.AND.T3.AND.T4.AND.
          $      .NOT.T5.AND.T6.AND.T7.AND.T8) AIN=0.1666666666666667D0
0209             IF(T1.AND.T2.AND.T3.AND.T4.AND.
          $      T5.AND..NOT.T6.AND.T7.AND.T8) AIN=0.1666666666666667D0
0210            ELSE
0211             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=(DN-6.D0*Q-9.D0)/DN7
0212             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=1.D0/DN7
0213             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=1.D0/DN7
0214             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND..NOT.T6.AND.T7.AND.T8) AIN=Q/DN7
0215             IF(T1.AND.T2.AND.T3.AND.T4
          $      .AND..NOT.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0216             IF(T1.AND.T2.AND.T3.AND..NOT.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0217             IF(T1.AND.T2.AND..NOT.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0218             IF(T1.AND..NOT.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.D0/DN7
0219             IF(T1.AND..NOT.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=(Q-1.D0)/DN7
0220             IF(.NOT.T1.AND.T2.AND.T3.AND.T4
          $      .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.D0/DN7
```

```
0221          IF(.NOT.T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=(Q-1.D0)/DN7
0222         END IF
      C
0223        ELSE IF(NS.EQ.7) THEN
0224         IF(TFULL) THEN
0225          IF(.NOT.T1.AND.T2.AND.T3.AND.T4.AND.
      $       T5.AND.T6.AND.T7.AND..NOT.T8) AIN=0.142857142857142800
0226          IF(T1.AND..NOT.T2.AND.T3.AND.T4.AND.
      $       T5.AND.T6.AND.T7.AND.T8) AIN=0.142857142857142800
0227          IF(T1.AND.T2.AND..NOT.T3.AND.T4.AND.
      $       T5.AND.T6.AND.T7.AND.T8) AIN=0.142857142857142800
0228          IF(T1.AND.T2.AND.T3.AND..NOT.T4.AND.
      $       T5.AND.T6.AND.T7.AND.T8) AIN=0.142857142857142800
0229          IF(T1.AND.T2.AND.T3.AND.T4.AND.
      $       .NOT.T5.AND.T6.AND.T7.AND.T8) AIN=0.142857142857142800
0230          IF(T1.AND.T2.AND.T3.AND.T4.AND.
      $       T5.AND..NOT.T6.AND.T7.AND.T8) AIN=0.142857142857142800
0231          IF(T1.AND.T2.AND.T3.AND.T4.AND.
      $       T5.AND.T6.AND..NOT.T7.AND.T8) AIN=0.142857142857142800
0232         ELSE
0233          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=(DN-7.D0*Q-8.D0)/DN7
0234          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=1.D0/DN7
0235          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=Q/DN7
0236          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND..NOT.T6.AND.T7.AND.T8) AIN=Q/DN7
0237          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND..NOT.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0238          IF(T1.AND.T2.AND.T3.AND..NOT.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0239          IF(T1.AND.T2.AND..NOT.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0240          IF(T1.AND..NOT.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0241          IF(.NOT.T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=1.D0/DN7
0242          IF(.NOT.T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=(Q-1.D0)/DN7
0243         END IF
      C
0244        ELSE IF(NS.GE.8) THEN
0245         IF(TFULL.AND.NS.EQ.8) THEN
0246          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=0.125D0
0247          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=0.125D0
0248          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND..NOT.T6.AND.T7.AND.T8) AIN=0.125D0
0249          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND..NOT.T5.AND.T6.AND.T7.AND.T8) AIN=0.125D0
```



```
0250          IF(T1.AND.T2.AND.T3.AND..NOT.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=0.125D0
0251          IF(T1.AND.T2.AND..NOT.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=0.125D0
0252          IF(T1.AND..NOT.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=0.125D0
0253          IF(.NOT.T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=0.125D0
0254         ELSE
0255          IF(T1.AND.T2.AND.T3.AND.T4.AND.T5
      $        .AND.T6.AND.T7.AND.T8) AIN=(DN-8.D0*Q-7.D0)/DN7
0256          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND..NOT.T8) AIN=Q/DN7
0257          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND..NOT.T7.AND.T8) AIN=Q/DN7
0258          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND..NOT.T6.AND.T7.AND.T8) AIN=Q/DN7
0259          IF(T1.AND.T2.AND.T3.AND.T4
      $        .AND..NOT.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0260          IF(T1.AND.T2.AND.T3.AND..NOT.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0261          IF(T1.AND.T2.AND..NOT.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0262          IF(T1.AND..NOT.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0263          IF(.NOT.T1.AND.T2.AND.T3.AND.T4
      $        .AND.T5.AND.T6.AND.T7.AND.T8) AIN=Q/DN7
0264         END IF
0265        END IF
0266       ELSE
0267        AIN=0.D0
0268       END IF
0269       RETURN
0270       END
```

```
      SUBROUTINE SORTJ(JAM,JSUB,M)
C
C SORT JAMMING EVENTS OF NON-SIGNAL CHANNELS ONLY
C
C TECHNIQUE: BUBBLE SORT, SINCE IT IS 7 ELEMENTS MAXIMUM
C           (THE FIRST ELEMENT, I.E. SIGNAL CHANNEL,
C           DOES NOT PARTICIPATE IN THE SORT; IT IS
C           JUST COPIED INTO OUTPUT ARRAY.)
C
C PROGRAMMER: R. N. FRENCH        DATE: 11 APR 84
C
      INTEGER*4 JAM(8),JSUB(8),JTEMP
      DO 1 I=1,M
      JSUB(I)=JAM(I)
1     CONTINUE
      IF(M.EQ.2) RETURN
      DO 10 I=2,M-1
C
C HERE WE WOULD NORMALLY USE THE STATEMENT
C
C
C     DO 20 J=I+1,M
C
C
C BUT IF WE DO, THE DEC F77 (V4.0) COMPILER GENERATES
C INCORRECT CODE AND THIS BECOMES AN INFINITE HANG-UP
C LOOP.  THIS IS DUE TO A FAILURE TO STORE UPDATED
C VALUE INTO J WHEN THE IF-CONDITION IS FALSE.
C THEN WHEN THE STORED VALUE IS FETCHED, IT IS ZERO
C AND THE LOOP JUST KEEPS ON GOING.  THUS THE
C LOOP WILL NEVER TERMINATE UNLESS THE IF CLAUSE
C JUST HAPPENS TO BE EXECUTED EVERY TIME THROUGH
C THE LOOP, I.E. THE INPUT ARRAY IS IN REVERSE ORDER.
C BY SIMULATING THE DO LOOP,
C WE BYPASS THE BUG IN THE COMPILER AND STORE J EVERY TIME.
C
      J=I+1
19    CONTINUE
      IF(JSUB(J).LT.JSUB(I)) THEN
        JTEMP=JSUB(I)
        JSUB(I)=JSUB(J)
        JSUB(J)=JTEMP
      END IF
C20   CONTINUE
C —
C — TERMINAL CODE FOR SIMULATED DO-LOOP (COMPILER BUG FIX)
C —
      J=J+1
      IF(J.LE.M) GOTO 19
10    CONTINUE
      RETURN
      END
```



```
      BLOCK DATA
C INITIALIZE SHARED CONSTANTS
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER*4 LOW,LINC,I200
      COMMON /SHARE2/ LOW(8),LINC(8),I200
      COMMON /GQWTS/ X(10),W(10)
      COMMON /ZETDEN/ DG(31),DSNR(5,4),DUMMY(89)
C DEFAULT LISTS FOR INTERACTIVE PARAMETER INPUTS ARE
C STORED (TEMPORARILY, UNTIL NO LONGER NEEDED) IN THE
C SAME AREA AS IS USED FOR SUBSEQUENT STORAGE OF THE
C PHASE-DIFFERENCE DENSITY
      DATA DG / .001D0, .01D0, .1D0, .5D0, 1.D0, 26*0.D0/
      DATA DSNR /13.3525D0, 12.3133D0, 10.9444D0, 0.D0, 0.D0,
     $           10.6065D0,  9.6284D0,  8.3524D0, 0.D0, 0.D0,
     $            9.0939D0,  8.1690D0,  6.9718D0, 0.D0, 0.D0,
     $            8.0783D0,  7.1996D0,  6.0696D0, 0.D0, 0.D0/
C ABSCISSAS (X) AND WEIGHTS (W) FOR
C 20-POINT GAUSSIAN QUADRATURE
      DATA X/ 0.07652652113349733375D0,
     1        0.22778585114164507808D0,
     1        0.37370608871541956067D0,
     1        0.51086700195082709800D0,
     1        0.63605368072651502545D0,
     1        0.74633190646015079261D0,
     1        0.83911697182221882339D0,
     1        0.91223442825132590586D0,
     1        0.96397192727791379126D0,
     1        0.99312859918509492478D0 /
      DATA W/ 0.15275338713072585069D0,
     1        0.14917298647260374678D0,
     1        0.14209610931838205132D0,
     1        0.13168863844917662689D0,
     1        0.11819453196151841731D0,
     1        0.10193011981724043503D0,
     1        0.08327674157670474872D0,
     1        0.06267204833410906357D0,
     1        0.04060142980038694133D0,
     1        0.01761400713915211831D0 /
C FREQUENTLY NEEDED CONSTANT ARRAYS AND SCALARS
      DATA LOW/8*0/,LINC/8*1/
      DATA I200/200/
      END
```

```
0001              SUBROUTINE DGAU20(A,B,F,ANSWER)
           C
           C 20-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL
           C
           C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4
           C
           C R. H. FRENCH, 21 JUNE 1983
           C
           C SHARING WEIGHT ARRAYS WITH THE ZETA-DENSITY INTEGRATOR
           C   --RHF, 16 APR 84
           C
0002              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003              COMMON /GQWTS/ X(10),W(10)
0004              ANSWER=0.D0
0005              BMAO2=(B-A)/2.D0
0006              BPAO2=(B+A)/2.D0
0007              DO 10 I=1,10
0008              C=X(I)*BMAO2
0009              Y1=BPAO2+C
0010              Y2=BPAO2-C
0011              ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0012       10     CONTINUE
0013              ANSWER=ANSWER*BMAO2
0014              RETURN
0015              END
```



```
0001              SUBROUTINE DBESJ(X,N,BJ,D,IER)
           C
           C        PURPOSE
           C           COMPUTE THE J BESSEL FUNCTION FOR A GIVEN
           C           ARGUMENT AND ORDER IN DOUBLE PRECISION
           C
           C        USAGE
           C           CALL DBESJ(X,N,BJ,D,IER)
           C
           C        DESCRIPTION OF PARAMETERS
           C           X  -THE ARGUMENT OF THE J BESSEL FUNCTION DESIRED
           C           N  -THE ORDER OF THE J BESSEL FUNCTION DESIRED
           C           BJ -THE RESULTANT J BESSEL FUNCTION
           C           D  -REQUIRED ACCURACY
           C           IER-RESULTANT ERROR CODE WHERE
           C                IER=0  NO ERROR
           C                IER=1  N IS NEGATIVE
           C                IER=2  X IS NEGATIVE
           C                IER=3  REQUIRED ACCURACY NOT OBTAINED
           C                IER=4  RANGE OF N COMPARED TO X NOT CORRECT
           C                       (SEE REMARKS)
           C
           C        REMARKS
           C         . N MUST BE GREATER THAN OR EQUAL TO ZERO, BUT IT
           C           MUST BE LESS THAN
           C                20+10*X-X** 2/3      FOR X  <= 15
           C                90+X/2               FOR X  >  15
           C
           C        METHOD
           C           RECURRENCE RELATION TECHNIQUE DESCRIBED BY H. GOLDSTEIN
           C           & R.M. THALER, 'RECURRENCE TECHNIQUES FOR THE
           C           CALCULATION OF BESSEL FUNCTIONS', M.T.A.C., V.13,
           C           PP.102-108 AND I.A. STEGUN AND M. ABRAMOWITZ,
           C           'GENERATION OF BESSEL FUNCTIONS ON HIGH SPEED
           C           COMPUTERS', M.T.A.C., V.11, 1957, PP.255-257
           C
           C      MODIFIED BY R.H. FRENCH, 1 JUNE 1983, TO HANDLE
           C          ZERO ARGUMENTS: J0(0) = 1.0, ELSE JN(0) = 0.0
           C
           C      MODIFIED BY R.H. FRENCH, 2 JUNE 1983, TO AVOID OVERFLOWS
           C          IN THE VICINITY OF CERTAIN ZEROS OF THE FUNCTION.
           C          THIS IS DONE BY SCALING DOWN ALPHA, FM, FM1, AND BJ
           C          (IN CASE IT WAS SAVED AS THE DESIRED ANSWER) IN THE
           C          EVENT THAT ALPHA BECOMES > 1.D28.  IF THIS HAPPENS,
           C          THE QUANTITIES LISTED ARE SCALED DOWN BY 1.D28.
           C      DOUBLE PRECISION VERSION, R.H. FRENCH, 13 OCT 1983
           C      MODIFIED BY R. H. FRENCH, 17 APR 1984, TO USE POLYNOMIAL
           C       APPROXIMATION FOR J0(X) IF X > 97.D0
           C
0002              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003              DIMENSION FWT(7),TWT(7)
```

```
              DATA FWT/ 0.79788456D0,-0.00000077D0,-0.00552740D0,
     $                 -0.00009512D0, 0.00137237D0,-0.00072805D0,
     $                  0.00014476D0/
              DATA TWT/-0.78539816D0,-0.04166397D0,-0.00003954D0,
     $                  0.00262573D0,-0.00054125D0,-0.00029333D0,
     $                 -0.00013558D0/
              BJ=.0D0
              IF(N)10,21,20
     10       IER=1
              RETURN
     20       IF(X)30,37,33
     21       IF(X)30,23,22
     22       IF(X-97.D0)33,33,300
     23       BJ=1.D0
              IER=0
              RETURN
     30       IER=2
              RETURN
     37       BJ=0.D0
              IER=0
              RETURN
     33       IF(X-15.D0)32,32,34
     32       NTEST=20.D0+10.D0*X-X**2/3.D0
              GO TO 36
     34       NTEST=90.D0+X/2.D0
     36       IF(M-NTEST)40,38,38
     38       IER=4
              RETURN
     40       IER=0
              N1=N+1
              BPREV=.0D0
     C
     C     COMPUTE STARTING VALUE OF M
     C
              IF(X-5.D0)50,60,60
     50       MA=X+6.D0
              GO TO 70
     60       MA=1.4D0*X+60.D0/X
     70       IFIXX=X
              MB=N+IFIXX/4+2
              MZERO=MAX0(MA,MB)
     C
     C     SET UPPER LIMIT OF M
     C
              MMAX=NTEST
     100      DO 190 M=MZERO,MMAX,3
     C
     C     SET F(M),F(M-1)
     C
              FM1=1.0D-28
              FM=.0D0
              ALPHA=.0D0
```



```
              IF(M-(M/2)*2)120,110,120
     110      JT=-1
              GO TO 130
     120      JT=1
     130      M2=M-2
              DO 160 K=1,M2
              MK=M-K
              BMK=2.D0*MK*FM1/X-FM
              FM=FM1
              FM1=BMK
              IF(MK-N-1)150,140,150
     140      BJ=BMK
     150      JT=-JT
              S=1+JT
              ALPHA=ALPHA+BMK*S
              IF(ALPHA-1.D28)160,155,155
     155      ALPHA=ALPHA/1.D28
              FM1=FM1/1.D28
              FM=FM/1.D28
              BJ=BJ/1.D28
     160      CONTINUE
              BMK=2.D0*FM1/X-FM
              IF(N)180,170,180
     170      BJ=BMK
     180      ALPHA=ALPHA+BMK
              BJ=BJ/ALPHA
              IF(DABS(BJ-BPREV)-DABS(D*BJ))200,200,190
     190      BPREV=BJ
              IER=3
     200      RETURN
     300      T=3.D0/X
              F0=FWT(1)+(FWT(2)+(FWT(3)+(FWT(4)+(FWT(5)+(FWT(6)+
     $        FWT(7)*T)*T)*T)*T)*T)*T
              TH=X+TWT(1)+(TWT(2)+(TWT(3)+(TWT(4)+(TWT(5)+(TWT(6)+
     $        TWT(7)*T)*T)*T)*T)*T)*T
              BJ=F0*DCOS(TH)/DSQRT(X)
              IF(D.LT.1.D-7)IER=3
              RETURN
              END
```

APPENDIX 8I

GENERALIZED Q-FUNCTION SUBPROGRAM

The following pages contain a listing of the FORTRAN-77 function subprogram which computes the generalized Q-function using Shnidman's algorithm [25]. This function is used by several of the programs contained in other appendices of this report.

```
PDP-11 FORTRAN-77 V4.0-1        15:49:25     7-Feb-84          Page 1

0001              DOUBLE PRECISION FUNCTION PNXY(N,X,Y,EPS)
           C SHNIDMAN'S GENERALIZED Q FUNCTION
           C REF.: DAVID A. SHNIDMAN, "EFFICIENT EVALUATION OF PROBABILITIES
           C          OF DETECTION AND THE GENERALIZED Q-FUNCTION," IEEE TRANS.
           C          ON INFO. THEORY, NOV. 1976, PP 746-751
0002              INTEGER*4 M
0003              DOUBLE PRECISION X,Y,EPS
0004              DOUBLE PRECISION LAMBDA,MLY,LY,LMF,LX,LRF,ANO2Y,E,XN,
             $OML,DEPS,P,B,R,XR,XRS,DY,YM,YMS,SUM,RD,RLX,FM
0005              DATA E/88.0296D0/
0006              IF(Y.EQ.0.D0)GOTO 2000
0007              DEPS=EPS
0008              DY=Y
0009              ANO2Y=N/(2.D0*DY)
0010              XN=X*N
0011              LAMBDA=1.D0-ANO2Y-DSQRT(ANO2Y*ANO2Y+XN/DY)
0012              OML=1.D0-LAMBDA
0013              IF(DLOG(DEPS).LT.-LAMBDA*DY+XN*LAMBDA/OML-N*DLOG(OML))
             $GOTO 100
0014              P=0.D0
0015              IF(Y.LT.N*(X+1.))P=1.D0
0016              PNXY=P
0017              RETURN
0018      100     B=N-1
0019              R=0.D0
0020              IF(E.LT.XN)GOTO 1100
0021              XR=DEXP(-XN)
0022      200     XRS=XR
0023              M=0
0024              IF(E.LT.DY)GOTO 800
0025              YM=DEXP(-DY)
0026      300     YMS=YM
0027      400     IF(B.EQ.M)GOTO 500
0028              M=M+1
0029              YM=YM*(Y/M)
0030              YMS=YMS+YM
0031              GOTO 400
0032      500     SUM=YMS
0033      600     R=R+1.D0
0034              M=M+1
0035              YM=YM*(Y/M)
0036      700     YMS=YMS+YM
0037              SUM=SUM+YM*(1.D0-XRS)
0038              XR=XR*(XN/R)
0039              XRS=XRS+XR
0040              IF(DEPS.LT.(1.D0-YMS)*(1.D0-XRS))GOTO 600
0041              P=SUM
0042              PNXY=P
0043              RETURN
0044      800     MLY=0.D0
0045              LY=DLOG(DY)
```

```
PDP-11 FORTRAN-77 V4.0-1        15:49:25     7-Feb-84          Page 2

0046      900     M=M+1
0047              FM=M
0048              LMF=DLOG(FM)
0049              MLY=MLY+LY-LMF
0050              IF(E.LT.DY-MLY)GOTO 900
0051              YM=DEXP(-(DY-MLY))
0052              IF(B.GE.M)GOTO 300
0053              YMS=YM
0054              SUM=0.D0
0055              RD=M-N
0056      1000    R=R+1.D0
0057              XR=XR*(XN/R)
0058              XRS=XRS+XR
0059              IF(RD.GT.R)GOTO 1000
0060              R=R+1.D0
0061              GOTO 700
0062      1100    RLX=0.D0
0063              LX=DLOG(XN)
0064      1200    R=R+1.D0
0065              LRF=DLOG(R)
0066              RLX=RLX+LX-LRF
0067              IF(E.LT.XN-RLX)GOTO 1200
0068              XR=DEXP(-(XN-RLX))
0069              B=B+R
0070              GOTO 200
0071      2000    PNXY=1.D0
0072              RETURN
0073              END
```

REFERENCES

[1] J.S. Lee, L.E. Miller, R.H. French, and Y.K. Kim, "Signal Design and Detection Strategies for LPI Communications in Electronic Warfare Environments," Final report on Contract N00014-81-C-0534, J. S. Lee Associates, Inc., May 1983 (AD-B073026L).

[2] I.S. Gradshteyn and I.M. Ryzhik, Table of Integrals, Series, and Products, Corrected and Enlarged Edition. New York: Academic Press, 1980.

[3] Peter Henrici, Applied and Computational Complex Analysis, Vol. 1. New York: John Wiley & Sons, 1974.

[4] Milton Abramowitz and Irene A. Stegun (eds.), Handbook of Mathematical Functions, National Bureau of Standards Applied Mathematics Series 55. Washington, D.C.: Government Printing Office, June 1964, Ninth printing, November 1970.

[5] William C. Lindsey and Marvin K. Simon, Telecommunication Systems Engineering. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1973.

[6] Anthony D. Whalen, Detection of Signals in Noise. New York: Academic Press, 1971.

[7] John M. Wozencraft and Irwin M. Jacobs, Principles of Communication Engineering. New York: John Wiley & Sons, 1965.

[8] Jhong S. Lee, Leonard E. Miller, and Young K. Kim, "Error Performance Analysis of Linear and Nonlinear Combining Square-Law Receivers for L-Hops Per Bit FH/BFSK Waveforms in Worst-Case Partial Band Jamming," Proc. 1983 IEEE Military Communications Conference (MILCOM '83), Vol. 1, Washington, D.C., Oct. 31-Nov. 2, 1983, pp. 22-28.

[9] Sam W. Houston, "Modulation Techniques for Communication, Part 1: Tone and Noise Jamming Performance of Spread Spectrum M-ary FSK and 2, 4-ary DPSK Waveforms," IEEE 1975 National Aerospace and Electronics Conference, June 10-12, 1975, pp. 51-58.

[10] S.L. Bernstein, "Error Rates for Square-Law Combining Receivers," Technical Note 1971-31, Massachusetts Institute of Technology, Lincoln Laboratory, 14 May 1971.

[11] Barry K. Levitt and Jim K. Omura, "Coding Tradeoffs for Improved Performance of FH/MFSK Systems in Partial Band Noise," National Telecommunications Conference 1981, pp. D9.1.1-D9.1.5.

[12] P.J. Crepeau and D.N. McGregor, "Anomalous Union Bound Behavior for MFSK Signaling on Inverse Linear Channels," IEEE Trans. on Comm., Vol. COM-31, No. 6, June 1983, pp. 848-849.

[13] Barry K. Levitt, "Use of Diversity to Improve FH/MFSK Performance in Worst Case Partial Band Noise and Multitone Jamming," 1982 IEEE Military Communications Conference, Vol. 2, October 17-20, 1982, pp. 28.2-1-28.2-5.

[14] Michael J. Massaro, "Error Performance of M-ary Noncoherent FSK in the Presence of CW Tone Interference," IEEE Trans. on Comm., Vol. COM-23, No. 11, November 1975, pp. 1367-1369.

[15] Gustav Doetsch, Guide to the Applications of the Laplace and Z-Transforms, 2nd English Edition. London: Van Nostrand Reinhold Company, 1971.

[16] Jan J. Tuma, Engineering Mathematics Handbook, First Edition. New York: McGraw-Hill Book Company, 1970.

[17] Seymour Stein and J. Jay Jones, Modern Communication Principles With Application to Digital Signaling. New York: McGraw-Hill Book Company, 1967.

[18] David Middleton, An Introduction to Statistical Communication Theory. New York: McGraw-Hill Book Company, 1960.

[19] Anonymous, Scientific Subroutines Programmer's Reference Manual, AA-1101D-TC. Marlboro, Mass.: Digital Equipment Corp., October 1981.

[20] Dave Howell et al. (eds.), 1984 IC Master, Vol. II. Garden City, N.Y.: Hearst Business Communications, Inc./UTP Division, 1984.

[21] Earl D. Rainville, Special Functions. Bronx, N.Y.: Chelsea Publishing Co., 1960.

[22] Athanasios Papoulis, The Fourier Integral and Its Applications. New York: McGraw-Hill Book Company, Inc., 1962.

[23] Athanasios Papoulis, Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill Book Company, 1965.

[24] Margaret Slack, "The Probability Distributions of Sinusoidal Oscillations Combined in Random Phase," Journal of the Institution of Electrical Engineers, Part 3, Vol. 93, March 1946, pp. 76-86.

[25] David A. Shnidman, "Efficient Evaluation of Probabilities of Detection and the Generalized Q-Function," IEEE Transactions on Information Theory, Vol. IT-22, No. 6, November 1976, pp. 746-751.

[26] A.J. Viterbi, "A Robust Ratio-Threshold Technique to Mitigate Tone and Partial Band Jamming in Coded MFSK Systems," 1982 IEEE Military Communications Conference, Vol. 2, October 17-20, 1982, pp. 22.4-1-22.4-5.

[27] K.S. Gong, "Performance of Diversity Combining Techniques for FH/MFSK in Worst Case Partial Band Noise and Multi-tone Jamming," Proceedings 1983 IEEE Military Communications Conference, Vol. 1, October 31-November 2, 1983, pp. 17-21.

END

FILMED

12-84

DTIC